Explorando el poder de Conditional Random Fields CRF's en la predicción de la estructura 3D de proteínas usando alfabetos estructurales

Trabajo de grado presentado por
Cristian Camilo Victoria Reyes

Universidad del Valle
Facultad de Ingeniería
Escuela de Ingeniería en Sistemas y Computación
Diciembre de 2016

Explorando el poder de los CRF's en la predicción de la estructura 3D de proteínas usando alfabetos estructurales

Estudiante
Cristian Camilo Victoria Reyes

Directora
Irene Tischer, Phd.

Universidad del Valle
Facultad de Ingeniería
Escuela de Ingeniería en Sistemas y Computación
Diciembre de 2016

Resumen

La función de las proteínas está dada por su estructura. Los métodos experimentales como cristalografía de rayos X y espectroscopia de resonancia magnética nuclear se usan para determinar la estructura de las proteínas. Estos tienen muchos problemas, son lentos, costosos y restringidos. Investigar métodos computacionales que hallen la estructura de una proteína dada su secuencia es esencial para resolver la brecha entre proteínas con estructura conocida y desconocida. La estructura global de las proteínas puede ser descrita por un conjunto de fragmentos estructurales cortos solapados, este conjunto se conoce como librería de alfabetos estructurales. Este proyecto se centra en el problema de predicción de la estructura local de las proteínas basado en su secuencia, alfabetos estructurales y características de las interacciones de los aminoácidos a largo rango.

Los predictores de estructura de proteínas normalmente basan sus predicciones en información evolutiva que relaciona la secuencia con la estructura. Estos funcionan muy bien en predecir estructura para proteínas con regiones conservadas en la secuencia, pero fallan para las proteínas con poca similitud y cuando tienen interacciones largas entre sus aminoácidos. Teniendo en cuenta lo anterior, este proyecto propone desarrollar un modelo gráfico no dirigido llamado Conditional Random Field para aplicarlo al problema de predicción de estructura local utilizando funciones características que captan información de las interacciones de los aminoácidos a largo rango (relación estructura-estructura).

Palabras clave: Conditional Random Filds (CRFs); Alfabeto estructural; Predicción estructura local; Relación estructura-estructura

Abstract

The protein function is given by its structure. Experimental methods as X-ray crystallography and nuclear magnetic resonance spectroscopy are used to determining protein structure. These have many problems, they are slow, expensive and restricted. Investigate computational methods to find the protein structure given its sequence is essential to solve the gap between proteins with known and unknown structure. The global protein structure can be described by a set of short overlapping structural fragments, this set is known as structural alphabet library. This project focuses on the problem of local structure prediction based on sequence information, structural alphabets and characteristics of the long range interactions of the protein amino acids.

Protein structure predictors typically base their predictions on evolutionary sequence information relating sequence-structure. They work very well predicting protein structure with conserved regions in the sequence, but fail for proteins with little sequence similarity and for the ones that have long range interactions between amino acids. This project proposes to develop a undirected graphical model called Conditional Random Fields and apply it to the problem of local structure prediction using features functions that capture information from the long range interactions of amino acids (structure-structure relationship).

Key words: Conditional Random Filds (CRFs); Structural Alphabet; Local Structure Prediction; Structure-Structure Relationship

Índice general

Lista de Abreviaciones

$SA_{10,3}$	Alfabeto estructura de 10 símbolos usando fragmentos de tres aminoácidos
CNF	Campo neuronal condicional
CRF	Campo aleatorio condicional
HMM	Modelo Oculto de Markov
PB	Bloques de proteínas
PDB	Banco de datos de proteínas
PSSM	Matrices de puntaje de posición específica
RMSD	Raíz de la media de los cuadrados de los errores
RMSDA	Raíz de la media de los cuadrados de los errores de valores angulares

Índice de tablas

Índice de figuras

Capítulo 1

Introducción

1.1. Descripción general

Las proteínas son moléculas formadas por secuencias de aminoácidos, que permiten que se plieguen en una forma particular 3D. No son moléculas rígidas; por el contrario, son dinámicas y pueden tener partes móviles acopladas a eventos químicos que les permite realizar tareas en los procesos dinámicos celulares. Algunas funcionalidades del vasto repertorio de las proteínas son: catalizadores, señalizar los receptores, interruptores, motores o bombas pequeñas [1]. En pocas palabras, las proteínas son la mano de obra molecular de los organismos vivos.

La proteínas han sido de gran interés investigativo por su importancia en el sector médico, biotecnológico y para la comprensión de la dinámica molecular biológica. Tienen una representación jerárquica de acuerdo a su estructura: La estructura primaria es la cadena lineal polipeptídica, representada como una secuencia sobre el alfabeto de veinte letras correspondiente a los aminoácidos. La estructura secundaria son las subestructuras locales recurrentes, las tres estructuras locales principales son: hélices (*a-helices*), hojas (*b-strands*) y giros (*coil*). La estructura terciaria es la estructura 3D de toda la proteína y se conoce como estructura nativa o estructura global de la proteína. Y la estructura cuaternaria es la conformación de varias proteínas que crean una más grande con una estructura definida, a cada cadena en la proteína se le llama una subunidad.

Existen métodos experimentales para hallar la estructura terciaria de la proteína, los más conocidos son cristalografía de rayos X y espectroscopia de resonancia magnética nuclear. Por medio de estos métodos, se han hallado las estructuras de una gran cantidad de proteínas; estructuras que son almacenadas en una base de datos llamada *Protein Data Bank* (PDB). Las proteínas resueltas en 3D, pueden ser consultadas en un formato de texto plano que contiene la posición 3D de cada átomo de los aminoácidos y un gran encabezado con información de la proteína. Toda la información almacenada en el PDB, sirve para tratar de entender cómo a partir de los aminoácidos la proteína se plega hasta obtener una estructura estable.

Los métodos experimentales para hallar la estructura terciaria de la proteína son laboriosos, costosos, y algunas veces no son factibles. Esto principalmente porque existen proteínas

que prosperan en ambientes difíciles de emular. Por otra parte, debido a la facilidad de los métodos para secuenciar ADN y obtener las secuencias de proteínas que codifican, se ha creado una enorme brecha entre las proteínas secuenciadas y las proteínas con estructura terciarias conocidas.

Muchos estudios han encontrado que la función de una proteína está dada por su estructura. Por eso, hallar la estructura de la proteína es clave para entender su función. Dada la gran cantidad de proteínas secuenciadas y los pocos, costosos y restringidos métodos de especificación de la estructura terciaria, surge la necesidad de desarrollar métodos computacionales factibles, que permitan encontrar la estructura terciaria y/o función de una proteína dada su secuencia. Siendo estos, objetivos primordiales en las investigaciones biológicas; esto básicamente porque ahorran los costos que originan los métodos experimentales y por su importancia en aplicaciones industriales y médicas.

Muchos trabajos biológicos se han centrado en la predicción y análisis de la estructura local de las proteínas dada su secuencia. Una de las formas es usar la representación de su estructura secundaria. Sin embargo, secuencias con estructuras secundarias similares, no tienen necesariamente la misma estructura global, y no proporciona mucha información para el modelamiento geométrico de las estructuras globales. Otra forma de analizar la estructura, es utilizar un conjunto de fragmentos representativos de las estructuras conocidas de las proteínas, llamados alfabetos estructurales (*structural alphabets*) o bloques básicos (*building blocks*), para tratar de describir las conformaciones estructurales locales de una forma más precisa. Predecir estructuras usando un ensamble 3D de alfabetos estructurales, ha proporcionado buenos resultados [22, 3, 24, 37], siendo esta estrategia un paso para la predicción de la estructura 3D global.

La librería I-Site, es una de las librerías de fragmentos estructurales más conocidas. Esta se obtiene formando grupos de fragmentos de estructuras conocidas, refinadas por medio de información estructural y alineamiento de fragmentos cortos. Para cada grupo, se encuentra una estructura 3D representativa [8]. Posteriormente, la librería se extiende y se utilizan modelos ocultos de markov (HMM), para capturar de una forma más compacta las relaciones entre motivos (secuencias cortas de aminoácidos, *motifs*) adyacentes. Resultando un modelo con muchas ramificaciones que sirve para predecir la estructura secundaria y los ángulos de torsión del *backbone* (Para cada aminoácido se emplea dos ángulos diedros φ, ψ para describir su correspondiente posición 3D) [9], ayudando a la predicción *ab initio* de la estructura de la proteína y a la detección de homólogos remotos.

El problema de la predicción de la estructura de las proteínas, ha sido estudiado por cinco décadas y todavía es un problema sin resolver. Este problema, se puede categorizar en dos enfoques. Primero, si existen proteínas similares en secuencia con estructuras conocidas, se usan métodos de comparación; entre estos están los basados en homólogos y en plantillas. Estos métodos se centran en buscar un modelo plantilla en las estructuras conocidas de las proteínas. Segundo, si no existen proteínas similares con estructuras conocidas, se tiene que encontrar la estructura usando sólo información de la secuencia. Estos métodos se agrupan en *ab initio* y *threading*. Los *ab initio* se basan en el supuesto que la estructura nativa de una proteína es la que tiene más baja energía. Estos se han estudiado, debido a que los métodos de comparación carecen de dar ideas del porque las proteínas toman una estructura, son ineficientes cuando no hay homólogos cercanos con estructura conocida, y cuando tienen homólogos pero difieren bastante en su estructura [26, 41]. Por otro lado, los métodos

de *threading* se basan en la idea de componer la estructura desconocida de una proteína usando segmentos de plegamientos locales conocidos de bases de datos, que relacionan secuencias con estructura. Este surgió porque para las tecnologías computacionales es muy difícil resolver los problemas de búsqueda planteados en *ab initio,* y los métodos de *threading* representan una pregunta más "fácil" de resolver.

Los resultados más importantes en predicción de la estructura de una proteína lo tienen arquitecturas en tuberías (*pipelines*), en donde se resuelven subproblemas hasta llegar a un modelo de la estructura. El servidor I-TASSER es clasificado como el mejor método para predicción 3D en la séptima evaluación del Critical Assessment of Structure Prediction (CASP), el cual usa métodos *threading* de reconocimiento de pliegues (*folds*) y luego usa métodos de *ab initio* para encontrar los modelos estructurales con mínima energía [73]. Es de gran interés científico, tecnológico e industrial contar con métodos computacionales eficientes que ayuden a entender cuáles son las normas que rigen el plegamiento de las proteínas, y predecir las estructuras 3D para crear aplicaciones médicas, biotecnológicas e industriales.

1.2. Problema

1.2.1. Descripción del problema

La predicción de la estructura de una proteína, es un problema difícil porque el plegamiento de la proteína es un proceso biológico complejo, donde los aminoácidos dependientes del contexto interactúan entre sí, formando una gran cantidad de estructuras 3D. Por medio de un conjunto de fragmentos estructurales cortos de proteínas, es posible modelar las estructuras globales de las proteínas de forma precisa [37]. Para hallar los fragmentos, se divide la estructura de las proteínas conocidas en fragmentos cortos, que son agrupados por similitud geométrica. Cada grupo está representado por una estructura, y estos son usados para construir o analizar las estructuras globales de las proteínas [22]. Una de las formas para predecir la estructura de una proteína, es usar la estrategia de ensamble de fragmentos estructurales cortos. Dada la secuencia de una proteína a la que se desea predecir su estructura, el problema consiste en asignar a los segmentos solapados de la secuencia de aminoácidos, uno o varios fragmentos estructurales.

Los conjuntos de fragmentos cortos representativos de las proteínas con estructura conocida, son llamados alfabetos estructurales. Cada fragmento estructural de un alfabeto se representa con un símbolo, lo que permite usarlos en métodos computacionales para analizar las estructuras de las proteínas [31]. Este trabajo se centra en el problema de predecir la estructura local usando un alfabeto estructural. Este problema es tan complejo como el de predecir la estructura global de una proteína, porque consiste en dividir la estructura global en partes más pequeñas, que son dependientes de muchas características biológicas. La importancia de las predicciones locales de las proteínas, radica en que estas predicciones pueden ser usadas en métodos de ensamble, que sirven como un primer paso para hallar la estructura global de las proteínas [22].

El problema de predecir la estructura local usando alfabetos estructurales y la información de las secuencias, ha sido estudiado usando enfoques probabilísticos y aprendizaje de

máquina [32]. Todos los métodos que predicen la estructura local, son enriquecidos por características de las proteínas para mejorar su tasa de predicción, dado que usar solo la secuencia de aminoácidos no proporciona buenos resultados. Para predecir, utilizan representaciones más genéricas que contienen información evolutiva de la relación secuencia-estructura. Los enfoques que tienen en cuenta las siguientes características mejoraron la tasa de predicción: información evolutiva, interacción entre aminoácidos a largo alcance o también llamada información estructura-estructura [19, 23], información de accesibilidad, y la estructura secundaria [42]. Aunque los métodos de predicción de la estructura local tienen una tasa de predicción baja para ser usados en aplicaciones, esta ha aumentado progresivamente desde los primeros trabajos. Joseph y de Brevern en [32], nombran algunas ideas de cómo mejorar la predicción de la estructura local. Ellos proponen usar información de accesibilidad ya que la estructura está influenciada por el ambiente y esta información mejora la predicción de estructura secundaria, incluir interacciones a largo alcance entre los aminoácidos (relación estructura-estructura), incluir señales de relaciones débiles de secuencia-estructura, y perfiles (*profiles*) dinámicos estructurales. Debido a la baja tasa de los métodos de predicción, es de interés explorar métodos para predecir la estructura local de las proteínas, para abstraer las características de las estructuras e incluir las características anteriormente mencionadas o explorar otras que sirvan para mejorar la predicción.

Los modelos discriminatorios son modelos condicionales que expresan la probabilidad de estados o etiquetas dadas las observaciones. En estos modelos, no es necesario modelar las dependencias de las observaciones. Debido a lo anterior, el modelo puede hacer uso de muchas características de una observación, como: diferentes niveles de granularidad de características para una observación, y/o características agregadas de las observaciones [38]. A diferencia de los modelos discriminatorios, los modelos generativos calculan la probabilidad conjunta de estados y observaciones; estos modelos tienen muchas ventajas pero también importantes limitaciones, como no poder representar múltiples características y dependencias complejas de las observaciones, porque modelar distribuciones sobre las observaciones, se vuelve complejo y el problema de inferencia se vuelve intratable; y no tenerlas en cuenta puede dirigir a reducir la precisión [38].

Un campo aleatorio condicional (*Conditional Random Field* CRF) propuesto por Lafferty *et al.* en [38], es un modelo probabilista discriminatorio. Este modelo expresa su probabilidad condicional por medio de funciones características, que son funciones que relacionan observaciones y etiquetas con un peso. Los CRFs superan a los modelos de markov de máxima entropía (MEMMs) y los modelos ocultos de markov (HMM), cuando los datos de entrenamiento tienen dependencias de alto orden [38]. Estos han sido aplicados a una gran cantidad de dominios, como por ejemplo: procesamiento del lenguaje natural, extracción de información, visión por computador, y bioinformática; obteniendo buenos resultados en todos [60]. En bioinformática se usó un CRF, para predecir pliegues beta-helix en proteínas [45], superando BetaWrap y HMMER, dos técnicas del estado del arte de predicción de pliegues.

Al momento de la elaboración de este trabajo, no se evidencia el uso de un CRF en la predicción de estructura local usando alfabetos estructurales. Sin embargo, la hipótesis es que los CRFs son efectivos para predicción de la estructura local de una proteína, porque proporcionan el poder de capturar patrones estructurales usando características

informativas locales, de largo rango, y conocimiento subyacente de los aminoácidos. En este trabajo se propone explorar el uso de los CRFs para predecir estructura local usando un alfabeto estructural dada la secuencia de aminoácidos de una proteína.

1.3. Objetivos

1.3.1. Objetivo General

Explorar el poder de los CRF's en la predicción de la estructura local de proteínas usando alfabetos estructurales.

1.3.2. Objetivos Específicos

1.3.2.1. Construir la base de datos a usar en la exploración

Conformar una base de datos que contenga proteínas del PDB lo suficientemente representativas estructuralmente, que esté anotada con un alfabeto estructural y que sirva para extraer información de las interacciones entre los aminoácidos a largo rango.

1.3.2.2. Modelar las funciones características a usar en el modelo

Definir las características de las interacciones entre los aminoácidos a largo rango en las proteínas que pueden mejorar la precisión de la predicción.

1.3.2.3. Construir el modelo CRF

Explorar configuraciones del modelo CRF que mejoren la precisión de la predicción con respecto a estructura local, y las interacciones entre los aminoácidos a largo rango.

1.3.2.4. Analizar y validar los resultados del modelo CRF

Definir los métodos para evaluar el poder de predicción y la capacidad de los modelos explorados para aplicarlos a los modelos desarrollados.

1.4. Alcance

Este proyecto obtendrá CRFs para predecir la estructura local de una proteína, usando un alfabeto estructural y características que captan información de las interacciones entre los aminoácidos. El objetivo de este proyecto, no es sobrepasar a los métodos del estado

del arte del problema de predicción de estructura local, sino explorar la capacidad de los CRFs para capturar interacciones entre los aminoácidos, y contrastar las razones por las que puede funcionar o no en este problema. Los CRFs no han sido usados para resolver el problema mencionado, por lo cual esta investigación generará un estado del arte sobre este modelo y la forma como se aplica para resolverlo. Además, fortalecerá las capacidades investigativas del autor. Aportará al avance en la línea de investigación del problema de predicción de estructura local, y a los esfuerzos del grupo de investigación en Bioinformática y Biocomputación de la Universidad del Valle en la investigación de las proteínas.

1.5. Justificación

Dado que la funcionalidad de una proteína depende de su estructura, y la gran brecha de proteínas con estructura conocida y desconocida (causada por las deficiencias de los métodos experimentales), es de gran interés científico crear métodos computacionales de predicción de estructura y función de proteínas, que ayuden a comprender el proceso de plegamiento. El conocimiento del proceso de plegamiento, la estructura, y la función de proteínas puede ser usado para desarrollar aplicaciones médicas, biotecnológicas e industriales. El problema de predicción de estructura local usando alfabetos estructurales, es importante porque es un paso que ayuda a comprender el proceso de plegamiento de las proteínas, y la predicción de la estructura 3D global; ya que los resultados de las predicciones pueden ser usados en métodos de ensamble para restringir el espacio de conformaciones estructurales. Debido a que las precisiones de los métodos actuales son muy bajas para que sean usados en la práctica, explorar nuevos métodos que mejoren la precisión es importante.

Predecir la estructura local es un proceso complejo, porque las estructuras locales son dependientes de las demás por las interacciones de las fuerzas moleculares y del contexto. Los predictores tratan de utilizar características de las proteínas, que le permitan abstraer conocimiento para mejorar la predicción, como: perfiles, interacción de alto rango entre los aminoácidos, accesibilidad y estructura secundaria. Los CRFs han sido aplicados en muchos dominios para segmentación, etiquetado de datos secuenciales y en problemas de clasificación con salida estructurada; pero nunca han sido usados para predecir estructura local. Este modelo, permite que pueda ser extendido para proporcionar el poder de capturar propiedades de la estructura de una proteína, y la habilidad de incorporar cualquier característica que pueda mejorar la precisión de la predicción. Las ventajas anteriores hacen atractivo la exploración de los CRFs en la predicción de estructura local, porque se adapta a la naturaleza de este problema y es un modelo más flexible comparado con otros modelos probabilísticos gráficos como HMM y MEMM.

Esta exploración es interesante porque contribuye a la formación e intereses académicos, brindando al investigador experiencia en el campo de aprendizaje de máquina. El tipo de modelo que se propone aquí para modelar la estructura global de una proteína, tiene un alto potencial en muchas áreas de aplicación. También contribuye a los esfuerzos del grupo de investigación en Bioinformática y Biocomputación de la Universidad del Valle en la investigación de las proteínas, el cual da respaldo que se lleve a cabo esta exploración, y por su importancia para la academia al evaluar el poder de este modelo que no se ha

usado antes en la predicción de la estructura local de las proteínas.

Capítulo 2

Marco Referencial

En este capítulo, se describe de forma concisa los conceptos y teorías relacionadas con la predicción de estructura de la proteína y el modelo gráfico campo aleatorio condicional CRF (Conditional Random Fields). Se divide en dos partes, la primera corresponde al marco teórico, que presenta los conceptos de predicción de estructura y estado del arte de la predicción de alfabetos estructurales. La segunda parte es el marco conceptual, que comprende el modelo gráfico CRF, específicamente su definición, comparación con los modelos gráficos dirigidos, la cadena CRF lineal, estimación de parámetros e inferencia, estado del arte de la aplicación de los CRFs lineales en biológica computacional, y una lista de software que implementan modelos CRFs.

2.1. Marco Teórico

2.1.1. Estructura de la proteína

Las proteínas son macromoléculas encargadas de realizar muchas funcionalidades en los seres vivos, están formadas por cadenas largas de aminoácidos. Un aminoácido es una molécula orgánica compuesta de un grupo amino (-NH2), un grupo carboxilo (-COOH), un hidrógeno, y una cadena lateral variable; todos los anteriores unidos a un carbono alfa. Es por medio de la cadena lateral, que se distingue un aminoácido de otro. Los aminoácidos están unidos por medio de enlaces peptídicos, la unión de los aminoácidos se da por medio de una reacción entre el grupo amino de uno y el carboxilo del otro, liberando una molécula de agua. La estructura de las proteínas se divide en cuatro niveles jerárquicos. La estructura primaria es la secuencia de aminoácidos. La estructura secundaria consiste de subestructuras locales del plegamiento de los aminoácidos. Hay tres tipos principales de estructuras locales recurrentes en las proteínas (otros autores subclasifican obteniendo 8 estructuras locales), clasificadas como: hélices (*a-helix*), que es una cadena de aminoácidos con estructura en forma de hélice, hojas (*b-strands*), que son dos cadenas de aminoácidos con estructura de dos láminas paralelas alineadas hacia la misma dirección u opuestas, y giros (*coil o loops*), que son cadenas de aminoácidos con estructura irregular. La estructura terciaria de la proteína se compone del plegamiento de la estructura secundaria tomando

una forma en 3D, que está dada por las fuerzas atómicas entre los aminoácidos que la conforman y el ambiente donde prosperan. Estas estructuras se representan mediante las coordenadas de los átomos de cada aminoácido. La estructura cuaternaria de la proteína se refiere a la unión de varias proteínas que se pliegan para formar una proteína más grande con estructura 3D definida, a cada proteína que la conforma se le llama una subunidad. La unión de proteínas se da cuando la superficie de una proteína puede interactuar con otra proteína a través de enlaces no covalentes, llamados sitios de enlace.

2.1.2. Protein Data Bank

El *Protein Data Bank,* es una base de datos que almacena la información de la estructura 3D de macromoléculas biológicas. Fue establecida en 1971 en Brookhaven National Laboratories (BNL), inicialmente contenía 7 estructuras resueltas, cada año empezó a aumentar las estructuras depositadas por las mejoras en las tecnologías de cristalografía, resonancia NMR y la necesidad de la comunidad científica de contar con un repositorio centralizado [4]. Al día de hoy, es la base de datos central que almacena las estructuras resueltas por métodos experimentales (con 104371 estructuras al 21 de Octubre del 2014).

2.1.3. Predicción de la estructura de la proteína

Por la gran importancia de la estructura de una proteína para conocer su funcionalidad, predecir la estructura 3D de la proteína dada su secuencia, es uno de los principales problemas a resolver en la bioinformática. Los métodos computacionales proporcionan una herramienta para llenar la brecha entre las proteínas con estructura conocida y desconocida, además son una alternativa a los costosos métodos experimentales que determinan la estructura de las proteínas. Los estudios de predicción de estructura de una proteína, se parten en tres enfoques descritos a continuación:

Comparativos

La forma más sencilla de hallar la estructura de una proteína desconocida, es comparar su secuencia con las proteínas de estructura conocida. Esto es posible, gracias a que la estructura se conserva en proteínas relacionadas evolutivamente. Los métodos de modelamiento comparativos (homólogos y plantillas), han tenido buenos resultados pero fallan cuando no existen homólogos cercanos, además carecen de dar conocimiento de cómo es el proceso de plegamiento de una proteína, y en no tener en cuenta que algunas proteínas se parecen en secuencia pero tienen diferentes estructuras globales.

Los métodos comparativos pueden tener buenos resultados al ser aplicados a proteínas que comparten más del 30 % de identidad en la secuencia con las proteínas de estructuras conocidas del PDB. Cuando una proteína tiene muchos homólogos cercanos, es probable que la estructura 3D sea similar a algún homólogo. Por el contrario, si una proteína no tiene tantos homólogos cercanos, su estructura puede o no ser similar con las estructuras de sus pocos homólogos. Y cuando una proteína no tiene homólogos cercanos su estructura es desconocida, pero puede coincidir con la de un homólogo remoto no detectado [32].

Ab initio

Los modelos que construyen la estructura 3D de una proteína desde cero, usando solo su secuencia, son conocidos como *ab initio*. Estos modelos se basan en la creencia que la estructura que asume una proteína es la que tiene menor energía libre. Estos métodos sirven para comprender los principios de cómo es el proceso de plegamiento de una proteína. Sin embargo, están limitados a resolver la estructura de pequeñas proteínas porque tienen que buscar la estructura en un gran espacio. La estructura de la proteína normalmente se busca en un espacio conformacional, por medio de una función de energía; las conformaciones finales son seleccionadas del espacio de búsqueda. Los métodos *ab initio* dependen entonces de la función de energía, de un método de búsqueda, y de un método de selección entre posibles estructuras [26][40].

Threading

Dado que hay superfamilias de proteínas que comparten un mismo pliegue, pero difieren mucho en sus secuencias, se crearon métodos de reconocimiento de pliegues. Se cree que la cantidad de diferentes pliegues tiene un límite y que las estructuras resueltas pueden cubrir la mayoría de estos. El objetivo de los métodos *threading* es asignar una estructura plantilla (pliegue) a la secuencia de una proteína con estructura desconocida. Es difícil capturar relaciones distantes entre secuencia y estructura por métodos de comparación y perfiles (*profiles*), por esto, los métodos *threading* tratan de expresar la relación entre la secuencia y una plantilla (pliegue) con la preferencia de los aminoácidos. Estos métodos, utilizan información adicional de la secuencia para detectar las relaciones distantes entre secuencia y estructura; como información evolutiva y restricciones conformacionales de la estructura de la proteína a identificar.

Hay dos estrategias principales usadas en *threading* para asignar compatibilidad entre secuencia y estructura. Los métodos globales buscan la plantilla para una secuencia de consulta que tenga asignado una posición a los residuos en el *backbone* de la plantilla y sean energéticamente compatibles. Los métodos locales, representan la relación secuencia-estructura por medio de la preferencia de cada aminoácido en un ambiente local de una estructura plantilla. La estrategia más común es predecir características estructurales locales, y luego unirlas para buscar la estructura global final [32].

2.1.4. Predicción con salida estructurada y predicción de la estructura de la proteína

La predicción con salida estructurada se refiere al problema de aplicación donde las variables observadas tienen una estructura secuencial o cualquier otra y las variables de salida (predichas) tienen estructuras complejas. Por ejemplo, en la predicción de la estructura de una proteína, se tiene como variables observadas la secuencia de aminoácidos y las variables de salida son las estructuras 3D. Otro ejemplo, es el etiquetamiento gramatical en procesamiento de lenguaje natural, se tiene una secuencia de palabras y se desea etiquetar a cada palabra con su correspondiente categoría gramatical. El problema con este tipo de predicción es que la salida tiene dependencias complejas.

En el aprendizaje supervisado, se tiene un conjunto de entrenamiento con su correspondiente observación x y salida y. El objetivo es aprender una función capaz de predecir el valor de salida a cualquier valor observado. Si y es discreto, el problema se conoce como clasificación y si es continuo como una regresión. En los problemas de clasificación con salida estructurada, la salida y puede ser un vector. Los escalares del vector no son independientes. Pueden estar asociados a otros por ubicación, por ejemplo y_i depende de y_{i+1}, y_{i+2}, o pueden tener otras relaciones más complejas. La esencia de la predicción estructurada es modelar las dependencias en un *framework* en lugar de tratarlas independientemente. Existen diferentes enfoques para modelarlos, pueden ser determinista o probabilista [43].

En este trabajo se abordará el problema de predicción de la estructura de la proteína desde el punto de vista probabilista con un modelo gráfico no dirigido descrito en el marco conceptual.

2.1.5. Contactos en las proteínas

Para analizar las proteínas y entender el proceso de plegamiento, se representa las proteínas de tal manera que sus características sean fáciles de observar y manipular. Una representación simple de la estructura, facilita diseñar algoritmos para predecir estructuras. Un ejemplo de representación, es la estructura secundaria, debido a su representación de tres estados usando símbolos. La predicción de estructura secundaria recibió mucha atención al inicio de la biocomputación.

Un mapa de contacto, representa la estructura de una proteína por medio de una matriz de booleanos de dos dimensiones. Cada dimensión corresponde a las posiciones de los aminoácidos de la proteína, y un valor es verdadero, si dos aminoácidos están más cerca que una distancia límite y falso en el caso contrario. Esta representación puede ser completamente transformada a la forma 3D original de la proteína, también es adecuada para aplicar algoritmos de minería de datos y aprendizaje de máquina; porque es independiente del marco de referencia de las coordenadas atómicas [71].

Existen varias formas de medir la distancia entre dos aminoácidos para generar mapas de contacto. La distancia entre carbonos beta (CB) con una distancia límite de 8 Angstrom fue escogida para usarse en los experimentos de la evaluación crítica de predicción de estructura (CASP) [71].

2.1.6. Alfabetos Estructurales

Los alfabetos estructurales o bloques básicos son conjuntos de fragmentos de estructuras representativas de las proteínas con estructura conocida, usados para representar toda o al menos la gran mayoría de las posibles estructuras del *backbone*. Estos tienen la capacidad de capturar la relación secuencia-estructura de las estructuras regulares frecuentes del *backbone*. La representación de las estructuras recurrentes por los alfabetos estructurales, representada en una dimensión con símbolos, permite usarlas en métodos computacionales para el análisis de las estructuras de las proteínas [31]. Estos han sido una alternativa al estudio de la estructura 3D, ya que enfoques como la estructura secundaria no proveen

descripción precisa a nivel 3D, porque omite la orientación relativa de la regiones conectadas. Además el estado giros de estructura secundaria representa el 50 % de todos los residuos, correspondientes a un gran conjunto de diferentes estructuras locales.

Los alfabetos estructurales se caracterizan por lo siguiente: número de fragmentos representativos, tamaño de los fragmentos, descriptores geométricos, y métodos de agrupamiento [3]. Todas estas características, se usan para generar alfabetos estructurales con propósitos en un problema específico [3]. Kolodny *et al.* en [37] mostró que la precisión en la descripción de las estructuras 3D por medio de los alfabetos estructurales depende de las características de complejidad (número de símbolos estructurales por aminoácido), tamaño de los fragmentos, y el número de fragmentos representativos.

Para describir la estructura 3D de una proteína se usa un método de solapamiento de elementos estructurales, que brinda una representación de una dimensión. Esto es, se asigna un elemento estructural a un fragmento de los aminoácidos basados en una medida de comparación con la estructura original. Esta descripción de estructura de una proteína puede realizarse con el objetivo de tener errores locales mínimos o teniendo un error global mínimo, y su calidad depende de la naturaleza del alfabeto estructural [37].

Se han desarrollado diferentes librerías optimizadas para la reconstrucción global, mientras otras se centran en especificidades de las secuencias de los fragmentos que son útiles para la predicción. Librerías con bastantes estructuras locales son útiles para describir las estructuras de forma precisa, y con limitadas estructuras son útiles para propósitos de predicción. Sin embargo, las librerías para predecir necesitan un equilibrio entre el número de estructuras locales para una correcta aproximación 3D y un número lo suficientemente pequeño para poder extraer relaciones secuencia-estructura [20].

2.1.7. Antecedentes de la predicción de la estructura local mediante alfabetos estructurales

La ausencia de similitud de estructura en secuencias similares, hizo que se tuvieran en cuenta otras representaciones para describir la estructura de una proteína, como es la estructura secundaria. Sin embargo, secuencias con estructura secundaria similar no necesariamente tienen la misma estructura 3D. Es por esta razón, que la representación de la estructura secundaria, no da mucha información de la conformación 3D del *backbone* de una proteína. Además, el estado *coil* o *loops* de la estructura secundaria tienen estructuras regulares con fuerte relación secuencia-estructura, como *hairpin loops*, *corner motifs*, *b-bulges*, etc. [32]. Y de esta forma, el *backbone* puede ser asociado por repetitivos elementos estructurales.

Muchos enfoques han sido desarrollados para predecir la estructura local basado en alfabetos estructurales, éstos están basados en métodos probabilísticos y aprendizaje de máquina [32]. A continuación se nombran algunos de los métodos de predicción; es importante tener en cuenta que los resultados de estos métodos usan diferentes conjuntos de prueba y criterios de evaluación de la predicción, por lo tanto no es adecuado comparar sus resultados.

El uso de modelos probabilísticos como HMM, han sido frecuentemente utilizados en bioinformática. Camproux *et al.* en [11], usaron un HMM para obtener un alfabeto estructural de 12 símbolos, independiente del conocimiento *a priori,* usando la información de las distancias de fragmentos solapados de cuatro carbonos alfa de un conjunto de secuencias. Se extendió posteriormente a un alfabeto de 27 símbolos, que es el tamaño más favorable de acuerdo con el criterio de información bayesiana, y permitió hacer análisis sobre la estructura local de las proteínas [10].

Sander *et al.* en [57], abordaron la pregunta: ¿dada la secuencia de un fragmento, que tanto se puede aprender acerca de la estructura que adopta? Para afrontarla utilizaron la idea de incorporar información estructural mientras hacen la partición del espacio de las secuencias. Definió un conjunto de 27 estructuras locales representativas de fragmentos de 7 aminoácidos utilizando una matriz de distancia de carbonos alfa para realizar el agrupamiento; este conjunto puede describir una proteína con un error de 1.19 Angstrom. Utilizó máquinas de soporte vectorial, árboles de decisión y bosques aleatorios (*Random forest*) para relacionar la secuencia con la estructura local. Realizó experimentos de predicción tomando como entrada a los clasificadores propiedades de los aminoácidos e información de perfiles de los aminoácidos utilizando la base de datos HSSP. Obtiene 23 %, 36 %, y 34 % de precisión para el árbol de decisión, la máquina de soporte vectorial y el bosque aleatorio.

Yang y Wang en [69], propusieron un método cuyo objetivo es predecir cuatro estados de la estructura local del *backbone* de un segmento de 9 aminoácidos, sólo con la información de la secuencia a consultar. Dada una secuencia de consulta, primero se predice la estructura secundaria, después se parte en segmentos solapados de 9 aminoácidos. Para cada segmento, el método consiste en seleccionar de LSBSP1 (base de datos de perfiles de secuencias de 9 aminoácidos basados en estructura local) los perfiles que tengan una similitud mayor a un umbral, y 60 % o más de la estructura secundaria correspondiente con la predicción. Posteriormente usando un método de consenso, la estructura local que se predice es el centro del grupo más grande de los perfiles.

El alfabeto estructural Protein Blocks (PB) [19], ha sido muy usado por su característica de alta predictibilidad, y en [34] lo caracterizan por ser muy informativo y adecuado para propósitos de predicción. Tiene 16 símbolos denotados por las letras $(a, b, \ldots, p)$, que se obtuvieron mediante un mapa autoorganizado (*self-organizing map*) teniendo en cuenta transiciones entre símbolos. Para capturar los 16 patrones estructurales, describieron el *backbone* de cinco aminoácidos con 8 ángulos diédricos asociados a los 5 carbonos alfa. La medida de similitud utilizada en el proceso de agrupamiento es la distancia euclidiana entre dos vectores angulares, llamada raíz de la media de los cuadrados de los errores de valores angulares (RMSDA por sus siglas en ingles). La codificación de la estructura de una proteína se lleva a cabo deslizando una ventana de 5 aminoácidos, cada fragmento es asignado con el PB más similar localmente a la estructura original de la ventana, de esta forma una proteína de tamaño L es descrita por $L-4$ fragmentos. Después de terminar el proceso de agrupamiento, se obtienen el primer resultado de los PB, pero este es refinado de la siguiente manera: Se codifica el conjunto de entrenamiento con los PB hallados, y se calcula una matriz de transición entre pares de PBs consecutivos que se transforman en frecuencias. Se recorre la base de datos de entrenamiento, se miran n PBs similares para cada fragmento leído y se selecciona el que tenga la frecuencia de transición más

alta con el previo PB, de esta forma fuerzan las transiciones entre PB cuando se observa similitud. Después realizan un procedimiento de disminución para seleccionar los PB que son lo suficientemente disimilares en estructura. Para lograrlo se inicia con b PBs, al final de cada ciclo en el agrupamiento, se prueban la similitud estructural y la similitud de transición entre dos PBs, y se elimina un PB considerado similar. El alfabeto PB puede ser caracterizado por su relación con la estructura secundaria, asignada usando STRIDE. Los PB a hasta c y d hasta f tienden a encontrase en hojas beta, los PB (k, l) y (n, o, p) tienden a encontrase en hélices alfa n-cap o c-cap respectivamente, los PB de g hasta j principalmente se encuentran en giros. Para predecir la estructura local (descrita por PBs) de una proteína dada su secuencia se usó la regla de Bayes. Su estrategia consiste en calcular $P(PB_k|X_k)$, la probabilidad del símbolo PB_k dada la información del fragmento X_k. Pero usa la tasa $\frac{P(X_k|PB_k)}{P(X_k)}$, deducida de la regla de Bayes, por su facilidad de cálculo por matrices de ocurrencia. Evalúan la estrategia anterior con Q_{16} y obtienen 30 % de precisión. La precisión es mejorada a 40.7 % con el concepto que un fragmento de secuencia puede estar asociado a diferentes PBs.

En [20] seleccionan las 72 secuencias más frecuentes de cinco PBs consecutivos no idénticas, que llaman Structural Words (SW). El número de SWs fue escogido porque es suficiente para codificar de forma 3D 85 % de los aminoácidos en una proteína, denominada cobertura. Los SW más frecuentes en los conjunto de datos están relacionados con las estructuras hélices alfa y hojas beta. La mayoría de los SW se sobreponen, los últimos 4 PBs de un SW son los primeros 4 PBs del siguiente SW. Por la propiedad anterior pueden definir cadenas largas y continuas, así que se usaron para crear una red. La red resume todos los caminos del *backbone* de las proteínas del conjunto de datos, no sólo describe las estructuras hélice alfa y hojas beta, también los giros, con una cobertura de 80 % usando sólo 58 SW. Prueban como los SW afectan la predicción, siguiendo la misma metodología de [19] para predecir PBs. Pero en lugar de predecir los 16 PBs, se predice los 72 SWs, y obtienen una tasa de predicción de 38 %.

La tasa de predicción del alfabeto PB fue mejorada a 39.9 % en [18], con un método llamado *pinning strategy*. Este método, usa la librería *Structural Words* (SW) que está compuesta por las 72 estructuras más frecuentes de 5 PBs consecutivos [20]. El método capturó la relación entre los SW y la secuencia, por medio de una matriz de ocurrencia de aminoácidos para cada SW. Dada la secuencia de una proteína con estructura desconocida, el método toma una ventana de un tamaño dado con un centro i que se desliza en un aminoácido. Para cada ventana se calcula el *adequacy score matrix* en las 72 matrices de ocurrencia de los SW. Después, para cada posición de los aminoácidos de la proteína de entrada se calcula el *score diversity index,* que representa las posiciones con mayor predictibilidad. Posteriormente, se recorre las posiciones ascendentemente basados en el *score diversity index*. El proceso de predicción inicia en la posición con menor *valor,* se toma el SW correspondiente con mayor *adequacy score*; esto para buscar los SW más probables para las posiciones adyacentes, llamados prefijo y sufijo. Se buscan el prefijo y el sufijo cuyos últimos y primeros 4 símbolos PBs correspondan a los del SW central, respectivamente. El prefijo y el sufijo se toman como predicciones si tienen el máximo *adequacy score* o si es mejor que un puntaje definido por el usuario. Cuando no es posible la extensión de un SW central, se prosigue con la siguiente posición según el *score diversity index*. Si la siguiente posición previamente ya tiene un prefijo o sufijo seleccionado en el proceso de extensión, se sigue con la siguiente posición. El proceso se repite hasta que se recorren

todas las posiciones en una proteína, obteniendo la predicción de PBs dada la secuencia. El método fue mejorado a 43.6 % teniendo en cuenta características de familias de proteínas, dado que muchas secuencias diferentes pueden pertenecer a la misma estructura local. El enfoque se basa en la proximidad de una ventana de una secuencia, a una ventana de una secuencia referente para cada familia de secuencias; similar al método de agrupamiento *k-means*. Para cada SW hallan *un número determinado de* familias de secuencias. Los resultados del método anterior dependen de la frecuencia de cada SW en el conjunto de entrenamiento.

Etchebest et. al en [24], basados en la metodología de predicción propuesta en [19] proponen un procedimiento de optimización estadístico para la relación secuencia-estructura, que mejora la tasa de predicción a 48.7 % Q_{16}. Además, verifican que las propiedades del alfabeto permanecen sin importar que sean evaluadas en un conjunto de datos más grande que el usado para hallarlo, también la distribución de la relación PBs y la asignación de estructura secundaria con STRIDE sigue igual a las halladas en [19]. La modificación de la predicción consiste en la mejora del procedimiento que relaciona n secuencias con un mismo PB de [19]. Es decir, el procedimiento que encuentra n familias de secuencias para un mismo PB. Este trabajo también es interesante porque evalúan si usar la predicción de estructura secundaria ayuda a mejorar la predicción, aunque los mejores resultados de predicción los obtienen combinando los resultados de PBs predichos con la predicción de estructura secundaria usando PSIPRED, esta mejora 1 %.

La tasa de precisión del alfabeto PB de 16 símbolos hallado en [19], fue mejorada a 51.2 %. Usando fragmentos de 7 símbolos estructurales (correspondientes a 11 aminoácidos) y un método de agrupamiento no supervisado llamado *hybrid protein model* (HPM). Obtuvo una librería de 120 prototipos solapados, donde cada grupo está representado por su prototipo de estructura media local, asegurando buena calidad de la aproximación estructural y capturando las características de largo rango de secuencia-estructura. El objetivo, fue predecir la estructura local usando la librería hallada y sus características estructurales; para lograrlo, se creó un experto para cada grupo usando una regresión logística. Una ventana de aminoácidos con estructura desconocida, se pasa como entrada a cada experto y cada uno da como salida la probabilidad que la ventana pertenezca al grupo respectivo, después se seleccionan las estructuras representativas de acuerdo con un umbral y la máxima cantidad de candidatos. Los resultados fueron evaluados por medio del criterio geométrico, una predicción es considerada correcta si el RMSD de los carbono alfa entre el mejor candidato y la estructura original es menor que 2.5 Å [3].

Posteriormente Bornot et al. en [7], proponen un nuevo esquema de predicción usando la librería de 120 prototipos solapados descrita en [3]. Cabe resaltar que es la librería más grande con respecto a la cantidad de aminoácidos de los prototipos usada para predecir, ya que anteriormente al 2006 se habían determinado alfabetos con fragmentos tan largos como 11 aminoácidos, pero nunca habían sido usados para predecir. El método de predicción se basa en un sistema experto entrenado para cada prototipo de la librería, se entrena el sistema para discriminar entre fragmentos de secuencias positivas (asociada al prototipo) y negativas (los otros prototipos). Cada experto arroja un puntaje de compatibilidad para una ventana de una secuencia. Los 120 puntajes son posteriormente ordenados, y finalmente se selecciona los 5 puntajes más altos. Los sistemas expertos son máquinas de soporte vectorial, como entrada a los expertos utilizan la matriz PSSM (*Position-Specific*

Scoring Matrix) que contiene información evolutiva. Evalúan las predicciones usando dos criterios: la predicción es considerada correcta cuando en la lista de los 5 candidatos contiene el prototipo original, y si la diferencia de RMSD entre carbonos alfa de por lo menos uno de los candidatos con el prototipo original es menor a 2.5 Å. Usando un conjunto de prueba obtienen 38.8 % y 63.1 % de precisión respectivamente. Además, analizan y comparan los resultados cuando usan máquinas de soporte vectorial o regresión logística en los sistemas expertos, tomando como entrada la secuencia y la matriz PSSM. Concluyen que la máquina de soporte vectorial y la regresión logística obtienen resultados similares cuando usan solo la secuencia como entrada. El uso de información evolutiva mejora significativamente los resultados de ambos, si se comparan con los resultados que usan sólo la secuencia. La máquina de soporte vectorial es superior con la matriz PSSM como entrada, contrario a lo anterior la regresión logística pierde rendimiento, básicamente este comportamiento recae sobre las propiedades de la regresión logística, y debido al pequeño conjunto de entrenamiento.

Ching-Tai Chen et. al en [12] proponen la herramienta HYPLOSP, método híbrido de predicción de estructura local por sus siglas en inglés. El método combina los resultados de un método basado en conocimiento y una red neuronal. Específicamente el método basado en conocimiento contiene información del alfabeto y estructura secundaria de secuencias de tamaño 7. Para entrenar la base de conocimiento utilizan un conjunto de datos codificado en un alfabeto estructural y su estructura secundaria asignada por DSSP. La base de conocimiento es extendida utilizando PSI-BLAST para encontrar homólogos remotos de proteínas con estructura conocida, recorren los resultados del alineamiento con una ventana de tamaño 7 que tengan 5 o más aminoácidos iguales y almacenan su información estructural. La predicción con la base de conocimiento se realiza en dos pasos: Paso 1, se hallan las secuencias homólogas con PSI-BLAST; Paso 2, se usan las secuencias similares de la base de conocimiento para votar por la estructura de cada aminoácido. La red neuronal tiene una capa oculta, se entrena con el perfil PSSM generado por la alineación y la asignación de estructura secundaria con DSSP. La predicción de la red toma como entrada el perfil y la predicción de estructura secundaria del servidor HYPROSP II. Para obtener la predicción final de un alfabeto estructural se combina los resultados de los métodos anteriores, simplemente se suma los dos puntajes de predicción correspondientes para cada posición, y se selecciona la estructura con mayor puntaje. Con tres alfabetos estructurales SAH, PB y STR, realizan validación cruzada de 10 iteraciones en un conjunto de 3,925 cadenas, sus resultados con la medida de evaluación Q_{16} para el alfabeto PB es de 59.54 % con la red neuronal, 57.79 % con la base de conocimiento y 63.24 % con HYPLOSP. Se evalúa la importancia del uso de la estructura secundaria en los métodos, si no la utilizan obtienen una disminución de 4.89 % para la red neuronal, 4.00 % para la base de conocimiento y 1.33 % para HYPLOSP. Además se explora por primera vez una red neuronal de dos etapas, la primera permanece igual a la anterior y la segunda toma los resultados de la primera con el fin de refinar la predicción teniendo en cuenta la relación estructura-estructura. La predicción de la red neuronal de dos etapas es 60.97 % y para HYPLOSP es 64.48 %. Cuando se pasa como entrada a la red neuronal la estructura secundaria asignada por DSSP en lugar de la predicha se tiene un límite superior de 69.14 %.

La tasa de predicción del alfabeto PB fue mejorada a 58.5 % Q_{16} usando redes neuronales artificiales (ANNs), y máquinas de soporte vectorial (SVM) de dos etapas, teniendo en cuenta la correlación entre la estructura local vecina, basada en la observación de corre-

lación entre estructuras secundarias, que produjo mejores resultados cuando se tomó en cuenta en los predictores de la estructura secundaria. El primer clasificador toma como entrada una PSSM con una ventana de tamaño w_1, la salida que produce es la probabilidad de la estructura del residuo del centro para cada símbolo del alfabeto. El segundo clasificador toma como entrada los resultados de la primera capa con una ventana w_2. La primera etapa se entiende como la abstracción secuencia-estructura; mientras que la segunda etapa, como la relación estructura-estructura. Usando dos alfabetos estructurales: DW y PB de 27 y 16 símbolos respectivamente, los clasificadores son entrenados con validación cruzada de 5 iteraciones, obteniendo los mejores resultados con PB por sus características, y su menor número de símbolos. Las redes neuronales para predecir PB tienen una capa oculta con 100 y 80 unidades ocultas, respectivamente. Algo interesante en estos resultados, es que la tasa mejora al entrenar usando un conjunto más grande, y las ANN toman menos tiempo de entrenamiento comparado con las SVM [23].

Dong *et. al* en [22] proponen un método para hallar la estructura local de una proteína y fragmentos de plegamiento con una librería de bloques básicos. Su método se basa en el concepto de jerarquía en el proceso de plegamiento de una proteína. Comienza con un proceso de entrenamiento, las proteínas con estructura conocida son partidas es trozos por un algoritmo llamado "Protein Peeling", con el obtienen los fragmentos iniciales. Los fragmentos son agrupados para producir una librería de bloques básicos y se construye un modelo bigrama de cualquier par de bloques básicos. Posteriormente se produce nuevos fragmentos en el conjunto de entrenamiento con el algoritmo "My-Peeling". El proceso anterior es repetido hasta que los fragmentos permanecen sin cambios. Los fragmentos están representados por estructura con ángulos diédricos y con el perfil de la secuencia. Las pruebas se realizan pasando un fragmento de entrada a una máquina de soporte vectorial (SVM) para obtener la asignación de los bloques básicos, después con el algoritmo de programación dinámica "My-Peeling" se construye los caminos posibles de plegamiento y se obtiene la predicción final. Construyen una librería con 180 bloques, que van de 4 a 7 en tamaño. Obtienen 61.4 % de precisión en la estructura local con el camino máximo de combinación retornado por "My-Peeling'.

Usando el alfabeto PB en una herramienta llamada Locustra que usa un esquema de predicción multiclase de dos capas de SVM, se reportó una tasa de predicción del 61 % Q_{16}. La primera capa toma como entrada un perfil de la secuencia a predecir. Esta capa usa el esquema de predicción multicapa de acoplamiento por pares: la clase positiva contiene muestras de una clase del alfabeto estructural, y la negativa de otra clase; para 16 símbolos estructurales se necesitan 120 clasificadores. La segunda capa, usa la salida de la primera como entrada y tiene un esquema de un clasificador por clase: la clase positiva contiene muestras de una clase del alfabeto estructural, y las negativas contiene muestras de las demás clases, obteniendo 16 clasificadores, uno por cada símbolo estructural [75].

La herramienta llamada svmPRAT para la anotación de residuos de proteínas, utiliza SVM para resolver el problema de predicción. Las características más importantes de svmPRAT son: la facilidad del usuario para proporcionar cualquier tipo de característica al predictor por medio de matrices, y el poder escoger dos tipos de funciones kernel: la estándar kernel de base radial, y una versión del kernel de base radial normalizada de segundo orden. Adicional a esto introduce un sistema de codificación de ventana para capturar señales para entrenar el modelo. Utiliza un *framework* en cascada de dos niveles, entrena dos

modelos L_1 y L_2. Ambos entrenan K uno-vs-el resto modelos de clasificación binaria para predecir una etiqueta. Los resultados del primer modelo son utilizados como entrada al segundo junto con las características de entrada para el segundo. svmPRAT fue usada para predecir la estructura local usando el alfabeto PB y obtuvo una tasa de precisión del 67 % Q_{16} en promedio con las siguientes variables: el perfil e información de la estructura secundaria con el servidor YASSPP como características de entrada, y el *kernel* de segundo orden [56].

2.2. Marco conceptual

2.2.1. Modelos Gráficos

Los modelos gráficos representan naturalmente distribuciones complejas sobre variables dependientes, como un producto de factores o subconjuntos de variables por medio de grafos. En estos, los vértices son variables aleatorias y las aristas representan relaciones probabilísticas directas entre las variables que se conectan. Representar gráficamente un modelo, provee la probabilidad conjunta y puede ser extendido, también se puede obtener conocimiento de las dependencias condicionales inspeccionando el grafo. Sus propiedades son utilizadas en los métodos de inferencia y aprendizaje [6]. Si los modelos gráficos representan las aristas con dirección, significa que es un modelo dirigido, llamado *Bayesian Networks* (BN), y si las aristas no tienen dirección, es un modelo no dirigido, llamado *Markov Network* (MN) o *Markov Random Fields* (MRFs).

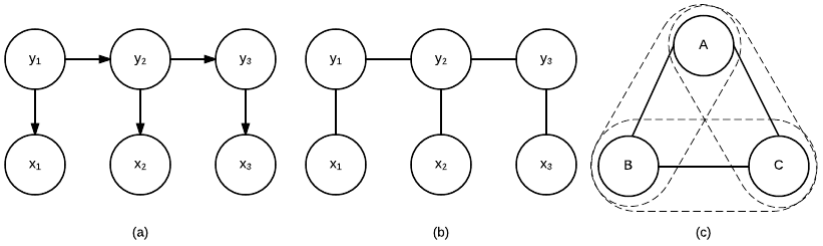


Figura 2.1: (a) Modelo gráfico dirigido de un HMM. (b) Modelo gráfico no dirigido MRF. (c) Los cliques máximos en el modelo gráfico no dirigido, son los pares {A,B} ,{B,C},{A,C}

Modelos dirigidos

Los modelos gráficos dirigidos, describen cómo una distribución de probabilidad se compone de distribuciones condicionales locales, ya que el grafo describe relaciones condicionales entre variables. La probabilidad conjunta se puede calcular como el producto de probabilidades condicionales de cada variable, condicionada en sus padres. Esto es, sea el grafo dirigido sin ciclos $G =< V, E >$, donde $pa(V_i)$ son los padres de la variable V_i en G. La probabilidad conjunta representada por el grafo es:

$$p(V) = \prod_{V_i \epsilon V} p(V_i \,|\, pa(V_i)) \tag{2.1}$$

La probabilidad $p(V_i|pa(V_i))$ es local, $pa(V_i)$ puede ser vacía si la variable no tiene padre.

Los modelos gráficos dirigidos más conocidos son bayesiano ingenuo y modelo oculto de Markov (HMM). Los HMM sirven para modelar datos secuenciales. La representación gráfica se muestra en la figura 2.1 (a). Dado un conjunto de observaciones $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ y un conjunto de clases o etiquetas ocultas que se quieren inferir $\mathbf{y} = \{y_1, y_2, \ldots, y_n\}$, se asume la propiedad de Markov de primer orden sobre los etiquetas, esto quiere decir que cada etiqueta y_t es independiente de los ancestros $y_1, y_2, \ldots, y_{t-2}$ dado el anterior y_{t-1} y que cada observación x_t es generada por el estado y_t. Dado lo anterior la probabilidad conjunta se expresa de la siguiente forma:

$$p(\mathbf{y}, \mathbf{x}) = \prod_{t=1}^{N} p(y_t \,|\, y_{t-1}) p(x_t \,|\, y_t) \tag{2.2}$$

Modelos no dirigidos

Los modelos gráficos no dirigidos son útiles para modelar fenómenos donde no se conoce la naturaleza de la dirección de las relaciones entre las variables. Estos también pueden expresar distribuciones de las variables en un grafo $G =< V, E >$ por medio de factores locales. La forma en que representan las distribuciones no es tan intuitiva como la de los modelos dirigidos, porque los factores locales no representan probabilidades condicionales, y por esto los resultados no son fácilmente interpretados. Los factores son funciones sobre un conjunto de variables, cuyos resultados son enteros positivos, estos subsumen la noción de probabilidad conjunta y probabilidad condicional [36]. De esta forma, pueden ser utilizados para representar probabilidades pero teniendo en cuenta que tienen que ser normalizados. Sea ψ un factor y C el conjunto de todos los cliques máximos del grafo G. $\Psi_c(\mathrm{V_c})$ es un factor sobre las variables del subconjunto $c \epsilon C$ y V_c es un vector de los valores de las variables. Este factor puede ser entendido como la compatibilidad entre los valores de sus variables de entrada. Entonces, se puede mostrar que la probabilidad conjunta del grafo no dirigido G puede ser factorizada de la siguiente forma:

$$p(V) = \frac{1}{Z} \prod_{c \epsilon C} \Psi_c(\mathrm{V_c}) \tag{2.3}$$

Donde Z es el factor de normalización, también conocido como la función de partición.

$$Z = \sum_{V} \prod_{c \epsilon C} \Psi_c(V_c) \tag{2.4}$$

Note que en la ecuación 2.4 la suma es sobre todos los valores de V. Esta función tiene una complejidad muy alta de cálculo cuando V tiene muchos valores.

La figura 2.1 (c) muestra los máximos cliques de un modelo gráfico no dirigido, la probabilidad de la familia de distribuciones que representa el gráfico puede expresarse como la multiplicación de estos factores, $p(A, B, C) = \frac{1}{Z} \psi_1(A, B) \psi_2(B, C) \psi_3(A, C)$ y Z es la suma de la productoria de los factores para todos los valores de A, B, C.

Los *Markov Network* (MN) o *Markov Random Field* (MRF) son modelos gráficos no dirigidos que representan relaciones de independencia condicional de una distribución. La figura 2.1 (b) muestra un *Markov Network* usado para datos secuenciales. Este tipo de modelos gráficos cumplen con la propiedad local de Markov, esto significa que un vértice y_t en un grafo G, es condicionalmente independiente de los demás vértices dado sus vecinos, implicando que sus vecinos aportan toda la información para predecir y_t [60]. La probabilidad conjunta del modelo gráfico de la figura 2.1 (b) sigue la forma de la ecuación 2.3, sólo hay que tener en cuenta que los factores representan $p(y_t \mid y_{N(t)})$ donde $N(t)$ retorna los índices de los vecinos para el vértice y_t.

Comparación modelos gráficos dirigidos y no dirigidos

La estructura en v de un modelo gráfico dirigido $A \rightarrow B \leftarrow C$ tiene las siguientes propiedades $A \perp C$ (A es independiente de C) y $A \not\perp C \mid B$ (A no es independiente de C dado B), resulta que no hay una estructura en un modelo gráfico no dirigido que pueda representar estas propiedades de independencia condicionales, porque los modelos gráficos no dirigidos son monotónicos y los dirigidos no son monotónicos, ya que al condicionar una propiedad de independencia esta puede no mantenerse. Dado que los modelos gráficos representan distribuciones de probabilidad usando las propiedades de independencia condicional del gráfico, los modelos gráficos dirigidos y no dirigidos representan conjuntos de distribuciones diferentes, pero existe una intersección entre estos conjuntos. La intersección corresponde a modelos gráficos que pueden ser representados de forma dirigida y no dirigida, donde sus propiedades condicionales son iguales. [51].

2.2.2. Comparación modelos generativos y discriminatorios

Aunque en ambos modelos se puede pasar de calcular la probabilidad conjunta a la condicional, usando el teorema de Bayes, existen muchas diferencias sutiles entre los dos modelos. Los modelos generativos describen como un vector de clases o etiquetas $\mathbf{y}$ puede probabilísticamente generar observaciones $\mathbf{x}$, calculando distribuciones de probabilidad de las observaciones, esto es $p(\mathbf{x} \mid \mathbf{y})$. Estos modelos calculan la probabilidad conjunta de estados y observaciones, describiendo una distribución de probabilidad sobre todas los posibles combinaciones entre estados y observaciones. Por lo anterior requieren representación de variables con pocos valores. Los HMM son modelos dirigidos generativos, tienen muchas ventajas pero también importantes limitaciones, las aplicaciones de HMM no representan múltiples características y dependencias complejas de las observaciones, porque modelar distribuciones sobre las observaciones se vuelve complejo y el problema de inferencia se vuelve intratable [38]. Pero ignorarlas puede reducir el rendimiento de los resultados del modelo [60].

Los modelos discriminatorios son modelos condicionales que expresan la probabilidad de clases o etiquetas dada las observaciones, esto es $p(\mathbf{y} \mid \mathbf{x})$, en los cuales no se desgasta en modelar las dependencias de las observaciones (la probabilidad no cuenta con las distribuciones condicionales de las observaciones $p(\mathbf{x})$). Además, el modelo puede hacer uso de muchas representaciones de características de una observación, como: diferentes niveles de granularidad y/o características agregadas de las observaciones. Las transiciones entre

estados o etiquetas, no sólo pueden depender del estado anterior sino del pasado o del futuro [38]. Los modelos de Markov de máxima entropía (MEMMs), son modelos gráficos dirigidos discriminatorios que tienen todas las ventajas mencionadas anteriormente, introducidos por [49]. Pero estos modelos, sufren del problema de sesgo de etiqueta [38], en donde la probabilidad de asignación de una etiqueta o estado depende sólo del estado anterior y la observación actual; esto significa, que están localmente normalizados y no permiten que una observación de un estado influencié a estados anteriores debido a su estructura en v [51].

Ambos modelos, se basan en calcular una distribución que involucra los vectores $(\mathbf{x}, \mathbf{y})$, pero lo hacen de formas distintas. Básicamente los generativos tienen en cuenta la probabilidad condicional de las observaciones $p(\mathbf{x} \mid \mathbf{y})$ y los discriminatorios no. Aunque usando la regla de Bayes, se puede pasar de calcular la probabilidad conjunta $p(\mathbf{y}, \mathbf{x})$ a obtener la condicional $p(\mathbf{y} \mid \mathbf{x})$, la forma como es calculada por el modelo discriminatorio permite modelar el problema basado en las dependencias de las clases o etiquetas y como éstas pueden depender de las observaciones.

2.2.3. Campos aleatorios condicional CRF (Conditional Random Fields)

Los CRFs son modelos de predicción de salida estructurada, donde la salida es un vector y sus elementos no son independientes. Han sido muy usados para la segmentación, etiquetado de datos secuenciales y en problemas de clasificación con salida estructurada como: reconocimiento de nombres de entidades y análisis sintáctico superficial en procesamiento del lenguaje natural, etiquetamiento de imágenes y reconocimiento de gestos en visión por computador, y predicción de genes en bioinformática, obteniendo buenos resultados [60]. Con los experimentos realizados en [38] se mostró que los CRFs superan a los MEMMs y HMM cuando los datos de entrenamiento tienen dependencias de alto orden.

Conditional Random Field (CRF) propuesto en [38], es un modelo probabilista discriminatorio que no sufre del problema sesgo de etiqueta. Es un MN donde los factores de los máximos cliques están condicionados a características de entrada. Considere un modelo gráfico no dirigido que usa vértices: X observaciones y Y etiquetas; un conjunto de factores $\psi = \{\psi_1(D_1), \ldots, \psi_m(D_m)\}$ que representan los máximos cliques del grafo; y una medida positiva no normalizada entre las variables Y e X dada por $\tilde{p}(Y, X)$, su distribución de probabilidad es:

$$p(Y \mid X) = \frac{1}{Z(X)} \tilde{p}(Y, X) \tag{2.5}$$

$$\tilde{p}(Y, X) = \prod_{i=1}^{m} \psi_i(Y_i, X_i) \tag{2.6}$$

$$Z(X) = \sum_{Y} \tilde{p}(Y, X) \tag{2.7}$$

Esta forma de representar un CRF es igual a la de un modelo no dirigido, sólo difieren

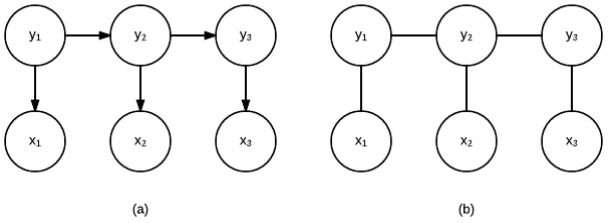


Figura 2.2: Estructura gráfica de (a) HMM; (b) Cadena lineal CRF.

en la forma como normalizan, ya que $Z(X)$ marginaliza sobre Y para obtener la probabilidad condicional $p(Y \mid X)$. Note que la normalización es diferente para cada conjunto de observaciones.

El poder de los CRFs radica en que no se codifica la distribución sobre variables de X, esto hace el modelo flexible y permite agregar ricas características de las variables observadas, cuyas dependencias pueden ser muy complejas o difíciles de comprender. De esta forma, se puede usar conocimiento del dominio del problema sin la necesidad de modelar sus distribuciones [36].

El modelo puede ser expresado de forma genérica log-lineal, si los factores son funciones lineales sobre funciones de activación conocidas como funciones características, que sirven para compartir pesos sobre todas las variables aleatorias. Estas funciones retornan 1 para una sola configuración de variables aleatorias, y 0 de forma contraria:

$$\psi_i(Y_i, X_i) = exp\left(\sum_{k,j} \theta^i_{kj} f^i_{kj}(Y_i, X_i)\right) \tag{2.8}$$

Los pesos θ^i y las funciones características f^i están indexadas por el factor i para indicar que los pesos pueden tener un conjunto de valores [60].

Los CRFs se conocen como una generalización de regresión logística, y su representación general es:

$$p(Y \mid X) = \frac{1}{Z(X)} \prod_{\psi_i \epsilon \psi} exp\left(\sum_{kj} \theta^i_{kj} f^i_{kj}(Y_i, X_i)\right) \tag{2.9}$$

2.2.3.1. Cadena lineal CRF

Los HMM tienen una relación análoga con una cadena lineal CRF. La figura 2.2 (a) muestra un HMM y la (b) su correspondiente modelo gráfico no dirigido. Los HMM representan la probabilidad conjunta expresada en la ecuación 2.2. Por la suposición de la propiedad de Markov, su probabilidad conjunta usa la probabilidad de transición entre una etiqueta dada la etiqueta anterior $p(y_t \mid y_{t-1})$ y la probabilidad de emisión $p(x_t \mid y_t)$. Esta probabilidad conjunta se puede expresar en una MN usando dos funciones características claves, una

para representar el clique de relación entre etiquetas, y otra para el clique de relación entre etiqueta-observación. Esto es:

$$p(Y,X) = \frac{1}{Z}\prod_{t=1}^{T} exp\left(\sum_{i,j}\theta_{ij}f_{ij}(Y_{t-1},Y_t) + \sum_{i,o}\alpha_{io}f_{io}(Y_t,X_t)\right) \tag{2.10}$$

La función característica para la relación entre etiquetas $f_{ij}(Y_{t-1},Y_t)$ se activa cuando $Y_{t-1} = j$ y $Y_t = i$, el peso es $\theta_{ij} = log(p(Y^{'} = j \,|\, Y = i))$ y para la emisión $f_{io}(Y_t,X_t)$ se activa cuando $Y_t = i$ y $X_t = o$, el peso es $\alpha_{io} = log(p(X = o \,|\, Y = i))$. La constante de normalización Z suma 1, $Z = 1$. Los índices i,j representan la etiqueta actual y la anterior respectivamente, el índice o representa la observación actual.

A partir de lo anterior se puede escribir la probabilidad condicional $p(Y \,|\, X)$ dada por el CRF de forma genérica, donde se itera las funciones características con los valores i, j, o representados por un k:

$$p(Y \,|\, X) = \frac{1}{Z(X)}\prod_{t=1}^{T} exp\left(\sum_{k=1}^{K}\theta_k f_k(Y_t,Y_{t-1},X_t)\right) \tag{2.11}$$

La cadena lineal CRF puede ser extendido, las funciones características pueden no sólo usar la observación actual, sino también características agregadas de las observaciones.

2.2.3.2. Inferencia en CRFs

El objetivo de la inferencia es calcular predicciones de las etiquetas $\mathbf{y}$ dado un conjunto de observaciones $\mathbf{x}$ para un modelo que ha sido entrenado y tiene unos parámetros θ. Hay dos problemas centrales en los modelos gráficos: calcular las distribuciones marginales $p(\mathbf{y}_a \,|\, \mathbf{x}; \theta)$ donde $\mathbf{y}_a$ es un subconjunto de valores de etiquetas. Normalmente el domino $\mathbf{y}_a$ de los valores consiste de una sola variable o un conjunto de variables vecinas. Y calcular las etiquetas $\mathbf{y}* = argmax_{\mathbf{y}}p(\mathbf{y} \,|\, \mathbf{x}; \theta)$, que son las más probables para una nueva observación $\mathbf{x}$ de entrada. En modelos con estructura de árbol, estas cantidades pueden ser calculadas de forma exacta, mientras en otros modelos generales comúnmente se recurre a aproximaciones. Contar con métodos de inferencia eficientes para CRFs es vital, porque son usados en el entrenamiento y la predicción. Para variables discretas los cálculos marginales pueden hacerse por sumas, pero el tiempo que requiere es exponencial con respecto al tamaño de $\mathbf{y}$. Ambos problemas de inferencia son intratables en grafos genéricos. Para el caso de las cadenas lineales CRFs, las tareas de inferencia pueden resolverse eficientemente con variantes de los algoritmos estándares de HMM, los cuales son *forward-backward* para calcular probabilidades marginales, y el algoritmo de Viterbi para calcular las etiquetas más probables [60].

Existen varios algoritmos para calcular inferencia exacta para modelos gráficos genéricos. Aunque estos modelos requieren tiempo exponencial en el peor de los casos, son buenos para resolver grafos que existen en la práctica. El algoritmo exacto más popular es descomposición en árbol (*junction tree*); sucesivamente agrupa variables hasta que el grafo

se convierte en un árbol. Una vez que se ha construido un árbol equivalente, se aplica un algoritmo para calcular la marginalización exacta en árboles. Un ejemplo de estos algoritmos, es propagación de creencias (*belief propagation*) que es una generalización del *forward-backward* para HMM.

Por la complejidad de la inferencia exacta, se han dedicado muchos esfuerzos en algoritmos de aproximación de inferencia. Dos clases han recibido la mayor atención: algoritmos Monte Carlo y variacionales. Los algoritmos Monte Carlo son algoritmos estocásticos que intentan mediante aproximaciones producir muestras de la distribución de interés. Los algoritmos variacionales son algoritmos que convierten el problema de inferencia en un problema de optimización, tratando de encontrar una aproximación simple más cercana a los marginales de interés. Estos algoritmos son más rápidos pero introducen sesgos comparados con los algoritmos Monte Carlo [60].

2.2.3.3. Estimación de parámetros en CRFs

La estimación de parámetros se refiere a calcular los parámetros $\theta = \{\theta_1 \dots \theta_k\}$ de un CRF. El caso más común es un aprendizaje supervisado, pero también existen estudios con aprendizaje semi supervisado en CRFs. Uno de los métodos para entrenar un CRF, es por medio de la máxima verosimilitud *(maximum likelihood)* que calcula los parámetros que hacen que los datos de entrenamiento tengan la probabilidad más alta en el modelo. El entrenamiento usado en este método es costoso, porque los CRF tienen muchas variables y una estructura más compleja comparada con otros clasificadores. En CRF con estructura de árbol, los parámetros que maximizan la verosimilitud se pueden encontrar por optimización numérica que usan los métodos de inferencia como subrutina. Los algoritmos de inferencia son utilizados para calcular la verosimilitud y sus gradientes. Afortunadamente la verosimilitud es una función convexa, lo que significa que se pueden usar procedimientos de optimización que probablemente converjan a la solución óptima [60].

Para CRFs con estructura general, el entrenamiento exacto por medio de máxima verosimilitud es intratable, entonces métodos de aproximación tienen que ser usados. De forma general, hay dos estrategias que tratan este problema. La primera es aproximar la verosimilitud por otra función más fácil de calcular, llamada verosimilitud sustituta; esta función es optimizada numéricamente. La segunda estrategia es usar un algoritmo de inferencia aproximado, que calcule las distribuciones marginales cuando la máxima verosimilitud las necesite.

2.2.3.4. Inferencia y estimación de parámetros en una cadena lineal CRF

Inferencia

Como se mencionó anteriormente, el problema de inferencia en cadenas lineales CRFs es igual al problema de inferencia de cualquier modelo gráfico, es así como los algoritmos estándares de inferencia de HMM se utilizan en cadenas lineales CRFs. El algoritmo de

Viterbi se utiliza para encontrar las etiquetas más probables dada una secuencia de observaciones y para hallar las probabilidades marginales se usa *forward* y *backward*; las probabilidades marginales usadas en la estimación de parámetros son las referentes a $p(y_t|x)$ conocida como marginal de los vértices y $p(y_t, y_{t-1}|x)$ como marginal de las aristas.

La idea de *forward* y *backward* viene de la necesidad de calcular la probabilidad de una observación $p(x)$. En [60] introducen una notación para simplificar las recursiones *forward* y *backward*. Acá nos basaremos en el tutorial de [60] para explicar las recursiones. Un HMM puede verse como un grafo de factores $p(y, x) = \prod_{t=1}^{T} \psi_t(y_t, y_{t-1}, x_t)$ donde $Z = 1$, y el factor se define como:

$$\psi_t(j, i, x) = p(y_t = j|y_{t-1} = i)p(x_t = x|y_t = j)$$

Una forma de calcular $p(x)$ es sumando sobre todas las posibles secuencias de etiquetas con tamaño T (número de observaciones), $p(x) = \sum_y \prod_{t=1}^{T} p(y_t, x_t)$. En [55] explican detalladamente las recursiones, mostrando que la complejidad de $p(x)$ es de $2TN^T$ porque para cada secuencia se hacen $2T$ operaciones (T operaciones para las transiciones y T operaciones para las emisiones) y hay N^Tsecuencias (N es el número de estados o etiquetas). Debido al orden exponencial, el cálculo de $p(x)$ no es factible incluso para valores pequeños de N y T. Pero, este problema puede ser resuelto con las recursiones *forward* y *backward* explicadas a continuación.

En [60] describen a $p(x)$ aplicando la ley distributiva de la siguiente manera:

$$p(x) = \sum_y \prod_{t=1}^{T} \psi_t(y_t, y_{t-1}, x_t)$$

$$p(x) = \sum_{y_T} \sum_{y_{T-1}} \psi_T(y_T, y_{T-1}, x_T) \sum_{y_{T-2}} \psi_{T-1}(y_{T-1}, y_{T-2}, x_{T-1}) \sum_{y_{T-3}} \psi_{T-2}(y_{T-2}, y_{T-3}, x_{T-2}) \dots$$

En lo anterior se observa que cada suma interna necesita información externa, y que esta es utilizada muchas veces durante el cálculo de la suma exterior. Es así como nace la idea de almacenar las sumas que se reutilizan para ahorrar trabajo. Se define una variable α_t llamada *forward* de tamaño N (N es el número de estados o etiquetas) que representa las sumas intermedias:

$$\alpha_t(j) = p(x_{<1,..,t>,}, y_t = j)$$

Se entiende, como la probabilidad conjunta de estar en el estado j en el tiempo t y observar $x_1, \dots, x_t$. Esta variable puede ser calculada de la siguiente manera:

$$\alpha_t(j) = \sum_{y_{<1,\ldots,t-1>}} \psi_t(j, y_{t-1}, x_t) \prod_{t'=1}^{t-1} \psi_{t'}(y_{t'}, y_{t'-1}, x_{t'})$$

Si se tienen todas las secuencias de tamaño $t-1$, calcular α_t es igual a multiplicar la probabilidad de cada secuencia por el factor ψ_t (la probabilidad de transición del ultimo estado de la secuencia a j, multiplicado por la probabilidad de emitir el símbolo x_t dado el estado j).

Se puede definir entonces la recursión:

$$\alpha_t(j) = \sum_{i \epsilon N} \psi_t(j, i, x_t)\, \alpha_{t-1}(i)$$

Dado que la probabilidad conjunta en el tiempo $t-1$ de estar en el estado i y observar $x_{<1,\ldots,t-1>}$ está dada por $\alpha_{t-1}(i)$, la probabilidad de pasar al estado j en el tiempo t será $\alpha_{t-1}(i)\psi_t(j, i, x_t)$. Sumando el producto anterior sobre todos los posibles estados N en el estado $t-1$, resulta la probabilidad del estado j en el tiempo t, $\alpha_t(j)$. La inicialización de la recursión es $\alpha_1(j) = \psi_0(j, y_0, x_1)$, el estado y_0 es un estado de inicialización y se entiende como $p(y_1 = j|y_0) = \pi(j)$, es decir la probabilidad inicial del estado j del HMM. Si se examina los cálculos necesarios para terminar la recursión, se necesitan el orden de TN^2 cálculos. Es claro que $p(x) = \sum_{y_T} \alpha_T(y_T)$.

De forma similar al *forward* se define la variable *backward* como:

$$\beta_t(i) = p\left(x_{<t+1,\ldots,T>} | y_t = i\right)$$

Se entiende como la probabilidad de ver las observaciones $x_{t+1,} \ldots, x_T$ dado que en el tiempo t el estado es i. La recursión se define como:

$$\beta_t(i) = \sum_{j \epsilon N} \psi_{t+1}(j, i, x_{t+1})\, \beta_{t+1}(j)$$

La recursión se inicializa con $\beta_T(i) = 1$. Para calcular la recursión se tienen en cuenta todas las posibles transiciones del estado j en la posición $t+1$, llevando acabo la transición entre el estado j al i y también la emisión de la observación x_{t+1} dado el estado j, posteriormente se tiene en cuenta las observaciones restantes con el término de la recursión $\beta_{t+1}(j)$. Por otra parte, se puede calcular $p(x)$ con *backward, así* $p(x) = \sum_{y_1} \beta_1(y_1)\, \psi_0(y_1, y_0, x_1)$.

Para calcular la probabilidad marginal $p(y_{t-1}, y_t|x)$, que es necesaria para la estimación de parámetros, se combinan las recursiones *forward* y *backward*. Hay dos maneras de llegar a la distribución de la probabilidad marginal, la primera es desde el punto de vista probabilista y el segundo es desde el punto de vista de factorización [60]. A continuación se explica desde el punto de vista probabilista:

$$p\left(y_{t-1}, y_t|x\right) = \frac{p\left(x|y_{t-1}, y_t\right) p\left(y_{t-1}, y_t\right)}{p\left(x\right)}$$

$$p\left(y_{t-1}, y_t|x\right) = \frac{p\left(x_{1,\ldots,t-1}, y_{t-1}\right) p\left(x_{t+1,\ldots,T}|y_t\right) p\left(y_t|y_{t-1}\right) p\left(x_t|y_t\right)}{p\left(x\right)}$$

$$p\left(y_{t-1}, y_t|x\right) = \frac{\alpha_{t-1}\left(y_{t-1}\right) \psi_t\left(y_{t-1}, y_t, x_t\right) \beta_t\left(y_t\right)}{p\left(x\right)}$$

En la parte dos se tuvo en cuenta las independencias condicionales de la cadena, las observaciones $x_{<1,\ldots t-1>}$ son independientes de $x_{<t+1,\ldots T>}$ y x_t dado y_{t-1}, y_t. El factor $p\left(x\right)$ actúa como normalizador y puede ser calculado por la recursión *forward* o *backward*.

Para calcular la secuencia de estados más probables dada la secuencia de observaciones $y* = argmax_y\, p\left(y|x\right)$, se utiliza el algoritmo de Viterbi. Este algoritmo usa la misma idea que la recursión *forward* pero en lugar de las sumas se utiliza la maximización:

$$\delta_t\left(j\right) = \max_{y_{<1,\ldots,t-1>}} \psi_t\left(j, y_{t-1}, x_t\right) \prod_{t'=1}^{t-1} \psi_{t'}(y_{t'}, y_{t'-1}, x_{t'})$$

Para calcularla, se toma la probabilidad máxima de obtener el estado j en el tiempo t a partir de la probabilidad máxima de cada estado anterior. Además se almacena el estado que maximizo la probabilidad, es decir el estado anterior que maximizó la probabilidad del estado actual. Posteriormente cuando se llega al final de la secuencia, se hace un retroceso para seleccionar la secuencia de estados que maximizan la probabilidad.

$$\delta_t\left(j\right) = \max_{i \epsilon N} \psi_t\left(j, i, x_t\right) \delta_{t-1}\left(i\right)$$

La recursión se inicializa con $\delta_1\left(j\right) = \psi_1\left(j, y_0, x_1\right)$. Al terminar la recursión se usa una recursión hacia atrás (o se memoriza el estado anterior para cada estado j) para hallar la secuencia de estados más probable:

$$y_T^* = \underset{i \epsilon N}{argmax}\, \delta_T\left(i\right)$$

$$y_t^* = \underset{i \epsilon N}{argmax}\, \psi_t\left(y_{t+1}^*, i, x_t\right) \delta_{t-1}\left(i\right) \quad para\ 1 < t < T$$

La generalización de los algoritmos anteriores (*forward*, *backward*, y Viterbi) a las cadenas lineales CRFs es simple. El *forward* y *backward* permanecen igual para las cadenas lineales CRFs, excepto que la definición del factor $\psi_t\left(y_{t-1}, y_t, x_t\right)$ es diferente al definido para el HMM. Se sabe que el modelo lineal CRF puede escribirse:

$$p(Y \mid X) = \frac{1}{Z(X)} \prod_{t=1}^{T} \psi_t \left(y_t, y_{t-1}, x_t\right)$$

$$\psi_t \left(y_{t-1}, y_t, x_t\right) = exp\left(\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, x_t)\right)$$

Con esa definición del factor, los algoritmos *forward*, *backward*, y Viterbi pueden usarse sin cambios. Pero la interpretación es diferente, para explicar las recursiones no se puede usar el punto de vista probabilista, estas tienen que ser definidas desde el punto de vista de la factorización, pero el resultado es el mismo a los definidos anteriormente [60]. También se usa $Z(x)$ en lugar de $p(x)$, y se halla de igual forma con *forward* o *backward* ($Z(x) = \sum_{i \epsilon N} \alpha_T(i)$, usando *forward*).

Las distribuciones marginales permanecen igual, solo cambia el factor de normalización.

$$p\left(y_{t-1}, y_t | x\right) = \frac{\alpha_{t-1}\left(y_{t-1}\right) \psi_t \left(y_{t-1}, y_t, x_t\right) \beta_t \left(y_t\right)}{Z\left(x\right)}$$

$$p\left(y_t | x\right) = \frac{\alpha_t\left(y_t\right) \beta_t \left(y_t\right)}{Z\left(x\right)}$$

Estimación de los parámetros

En esta sección se explica como los modelos lineales CRFs en este trabajo son entrenados, basado en el tutorial [60]. El objetivo de la estimación de parámetros es obtener los parámetros $\theta = \{\theta_1 \ldots \theta_k\}$ a partir de un conjunto de datos etiquetados. Una forma de entrenar una cadena lineal CRF es por máxima verosimilitud, esto es, los parámetros se seleccionan de tal manera que el conjunto de entrenamiento tiene la máxima probabilidad bajo el modelo. Es importante saber que la verosimilitud para las cadenas lineales CRFs es una función convexa de los parámetros, lo que significa que existen poderosos procedimientos de optimización numérica disponibles, que pueden converger a la solución óptima.

Dado un conjunto de entrenamiento $D = \left\{x^{(i)}, y^{(i)}\right\}_{i=1}^{N}$, donde cada $x^{(i)} = \left\{x_1^{(i)}, x_2^{(i)}, \ldots, x_T^{(i)}\right\}$ es una secuencia de entrada y cada $y^{(i)} = \left\{y_1^{(i)}, y_2^{(i)}, \ldots, y_T^{(i)}\right\}$ es una secuencia de etiquetas. Para simplificar, se asume que todas las secuencias tienen el mismo tamaño T. Para estimar los parámetros típicamente se usa máxima verosimilitud penalizada (con regularización), el *log* de la verosimilitud es apropiado para la estimación.

$$\ell\left(\theta\right) = \sum_{i=1}^{N} log\, p\left(y_i | x_i; \theta\right)$$

Para calcular el estimado de la máxima verosimilitud, se maximiza $\ell\left(\theta\right)$, esto es $\hat{\theta} = sup_\theta\, \ell\left(\theta\right)$.

Después de sustituir el CRF lineal en la *log* verosimilitud se obtiene:

$$\ell(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K}\theta_k f_k\left(y_t^i, y_{t-1}^i, x_t^i\right) - \sum_{i=1}^{N} log\, Z\left(x^i\right)$$

Muy a menudo los modelos tienen muchos parámetros, en tales casos para reducir el sobre entrenamiento es necesario utilizar regularización, que penaliza valores de vectores cuya norma es muy grade. Una opción común de penalización es la basada en la norma euclidiana de θ y en un parámetro de regularización $\frac{1}{2\sigma^2}$ que determina la fuerza de la penalización. Entonces la *log* verosimilitud regularizada L2 es:

$$\ell(\theta) = \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{k=1}^{K}\theta_k f_k\left(y_t^i, y_{t-1}^i, x_t^i\right) - \sum_{i=1}^{N} log\, Z\left(x^i\right) - \sum_{k=1}^{K}\frac{\theta^2}{2\sigma^2}$$

El parámetro σ^2 es libre, y determina cuanto penalizar pesos grandes. La idea es reducir el potencial de una pequeña cantidad de características que dominan la predicción.

En general la *log* verosimilitud no puede ser maximizada con métodos analíticos, por esto se tienen que utilizar métodos de optimización numérica. Las derivadas parciales de la *log* verosimilitud son:

$$\frac{\partial \ell}{\partial \theta_k} = \sum_{i=1}^{N}\sum_{t=1}^{T} f_k\left(y_t^i, y_{t-1}^i, x_t^i\right) - \sum_{i=1}^{N}\sum_{t=1}^{T}\sum_{y,y'} f_k\left(y, y', x_t^i\right) p\left(y, y' | x_t^i\right) - \frac{\theta_k}{\sigma^2}$$

En el tutorial [60] explican una interpretación de la derivada, expresan que el segundo término derivado de $log\, Z\,(x)$ se entiende como el valor esperado de la distribución empírica del primer término. Un hecho interesante es que cuando no hay factor de normalización la derivada es cero cuando ambas distribuciones son iguales, y lo anterior es un hecho de la verosimilitud en distribuciones de la familia exponencial.

Para calcular la *log* verosimilitud y sus derivadas es necesario usar la inferencia. La verosimilitud necesita calcular $Z\left(x^i\right)$, que es la suma de todos los posibles estados, además para calcular las derivadas se necesita calcular la probabilidad marginal $p\left(y, y' | x_t^i\right)$. Ambas tareas dependen de las observaciones x^i, por lo tanto en la función de verosimilitud para cada secuencia de observaciones se necesita calcular la inferencia. De forma contraria, los métodos generativos, como los modelos no dirigidos generativos pueden ser entrenados por máxima verosimilitud, pero su factor de normalización sólo depende de los parámetros, no de ambos (los parámetros y la observación). Debido a la cantidad de tareas de inferencia que se necesitan para estimar los parámetros con máxima verosimilitud, se han utilizado otros enfoques como gradiente ascendente aleatorio.

En este trabajo, para hallar los parámetros se utiliza la versión de memoria limitada de BFGS. Esta técnica ha sido muy utilizada en diferentes trabajos, mostrando buen rendimiento. Y es preferida porque converge más rápido que los enfoques de escalamiento iterativos [38], ya que utiliza información de la curvatura con la hesiana aproximada (matriz

de segundas derivadas). Se usa una aproximación porque es una matriz cuadrada en el número de parámetros.

2.2.3.5. Extensiones de CRFs

Los CRFs son modelos flexibles que han sido adaptados para solucionar diferentes tipos de problemas donde los CRFs convencionales presentan dificultades. Muchos de los problemas en donde se pueda usar un CRF requieren modelos gráficos más complejos que los convencionales, es así como han nacido muchas adaptaciones a los CRFs. Un ejemplo de extensión son los CRFs dinámicos [61]. A continuación se listan otras:

Semi-Markov Conditional Random Fields

Los Semi-Markov CRF son una generalización de los CRFs secuenciales. Para una secuencia de entrada $\mathbf{x}$ da como salida una segmentación de $\mathbf{x}$, en el que las etiquetas son asignadas a segmentos de $\mathbf{x}$ en lugar de ser asignadas a cada elemento x_i de $\mathbf{x}$. Lo anterior se refiere a que una etiqueta s_i persiste para un tiempo d_i sin unidad. Después que el tiempo ha pasado, el sistema pasa a una nueva etiqueta $s^{'}$ que depende solo de s_i. Las ventajas de este modelo, son que las características pueden medir propiedades de los segmentos y no hay transiciones markovianas en los elementos de un segmento. A pesar de su poder adicional, la complejidad para aprender e implementar inferencia exacta son polinomiales, a menudo un factor constante menor al de los CRFs. Este modelo presenta mejoras en la tarea de extracción de información comparado con cadenas lineales CRFs [58].

Hidden Conditional Random Fields

Los HCRFs son extensiones de CRFs en donde se agregan variables ocultas entre las observaciones y la variable de las etiquetas. Las variables ocultas captan cierta estructura escondida de cada etiqueta, y contribuyen a mejorar el reconocimiento de patrones de objetos en imágenes y reconocimiento de gestos [66]. Los HCRFs superaron el estado del arte en su momento, en el reconocimiento de patrones debido a que los CRFs no calculan las probabilidades de las observaciones.

Deep-Structured Conditional Random Fields

Los CRFs lineales modelan restricciones entre saltos de etiquetas por medio de restricciones entre etiquetas consecutivas. CRFs más complejos pueden modelar restricciones de alto orden entre etiquetas directamente. En algunos problemas modelar estas relaciones de alto rango entre etiquetas puede mejorar el rendimiento, pero a un costo muy grande, donde la complejidad de estimación de parámetros e inferencia es sustancialmente alta y se tienen que usar métodos de aproximación para hacerlos tratables. La estrategia de un deep-structured CRF, consiste en usar múltiples capas de CRFs menos complejos, como los CRFs lineales (en lugar de usar CRFs complejos), para aproximar las relaciones de largo

rango entre estados sin sacrificar la eficiencia. Las observaciones de cada capa consisten de la secuencia de observaciones de la capa anterior y el resultado de las probabilidades marginales posteriores del fragmento. Ya que las características en un CRFs pueden ser construidas usando toda la secuencia de observaciones, las características de las capas altas pueden ser construidos usando las creencias de las capas bajas de los fragmentos, que están lejos de un fragmento actual [70].

Conditional Neural Fields

Los campos neuronales condicionales son una extensión de las cadenas CRFs lineales que hace una transformación de las observaciones con una red neuronal de una capa oculta. La red neuronal para cada instante en la secuencia de las observaciones es la misma. En lugar de pasar una ventana de observaciones directamente a las funciones características del CRF lineal, la red neuronal toma una ventana y produce una representación de las observaciones que es linealmente separable por el CRF. Sus aplicaciones se centran en problemas donde los CRF lineales por ser un modelo lineal, no es capaz de discriminar correctamente dadas las observaciones crudas de los datos. Como es el caso del reconocimiento de escritura y la predicción de estructura secundaria [53].

Otras extensiones

Dado que los CRF son buenos en aplicaciones donde se necesite modelar relaciones entre variables, pero empíricamente se ha mostrado que son malos en problemas donde las observaciones no pueden ser linealmente separables [64]. Hasta ahora, se están combinando modelos buenos para aprender representaciones distribuidas (*distributed representations*), también conocido como modelos de aprendizaje profundo (*deep learning*) [39], con modelos gráficos como los CRF. Un ejemplo es la combinación de una cadena CRF lineal con una Red Neuronal Recurrente (LSTM), este modelo se evalúa en la tareas de etiquetamiento gramatical, análisis sintáctico superficial y reconocimiento de entidades, produciendo el estado del arte (o cerca) en las tareas anteriores [28].

2.2.4. Aplicaciones de Cadenas de Campos aleatorios condicionales lineales en biología computacional

Las cadenas lineales CRFs han sido aplicadas exitosamente a problemas de etiquetamiento de datos secuenciales. Liu *et al.* en [44] estudiaron diferentes modelos gráficos para predecir estructura secundaria, pero como métodos de combinación. Esto significa que los resultados de uno o varios predictores de estructura secundaria se toman como observaciones en un modelo gráfico, bajo la restricción del tamaño de la secuencia. El problema se conoce como un problema de consenso, donde las observaciones del modelo son las etiquetas predichas o los puntajes de la estructura secundaria, y estas se combinan para obtener la predicción de las etiquetas finales. En sus experimentos se encontró que las observaciones de puntajes, brindan más información que las etiquetas, lo que mejora las predicciones. También, las cadenas CRFs son los mejores modelos gráficos y sobrepasan los resultados de los modelos

basados en ventana, porque aprovechan de forma directa las relaciones entre etiquetas vecinas. Además, los CRFs mejoraron la predicción de hélices alfa y hojas beta.

Lukov *et al.* en [46] evaluaron la capacidad de la cadena lineal CRF para predecir estructura secundaria, a partir de los aminoácidos y sus características. Por medio de un CRF aprenden las relaciones para predecir la estructura secundaria hélice alfa en proteínas de membrana. El problema se plantea con la secuencia de observaciones de aminoácidos representados por x, y las etiquetas representadas por y. Los valores de las etiquetas son 1 o 0 que denota la presencia o ausencia de la estructura hélice alfa en una posición. Basados en información específica del problema, evaluaron diecisiete características de los aminoácidos que ayudan a diferenciar las estructuras hélice, algunas son: probabilidad de un aminoácido de estar en estructura hélice en membranas, hidrofobicidad de un segmento, puntaje hidrofílico en un segmento, clasificación en nueve grupos de las propiedades de los aminoácidos. Encontraron que los valores de hidrofobicidad son buenos para localizar estructuras hélice, y sus resultados soportan las observaciones que la estructura hélice está influenciada por los aminoácidos vecinos, y que no interesa su orden. Obtienen una precisión de 84 % Q_2 sobrepasando veintiocho métodos diferentes, concluyen que los CRF son buenos para integrar información diversa, no independiente, y de bajo nivel para hacer predicciones a alto nivel.

Zhang et al. en [72] aplicaron una cadena lineal CRF en la predicción de ángulos diédricos del *backbone*. El problema consiste en predecir los ángulos diédricos $\Psi_k = (\varphi_k, \psi_k)$ para cada aminoácido A_k de una proteína con tamaño n usando solo la información de los aminoácidos. Ya que los valores de estos ángulos inicialmente están entre -180 hasta 180 grados, son discretizados de la siguiente manera: primero, el ángulo ψ es desplazado -50 grados, formando una distribución de dos picos; segundo, los ángulos son divididos en 18 partes, cada parte corresponde a una diferencia de ángulos de 20 grados. Se le asigna un número a cada parte para predecir. Evalúan el modelo utilizando Q10 %, cada ángulo es predicho con el valor medio de la parte y se toma como verdadero positivo cuando su valor esta hasta 36 grados del valor experimental. Obtienen 87 % y 84 % de precisión en 10 iteraciones de validación cruzada para φ y ψ, respectivamente. Analizan las fuentes de error de la predicción usando la composición de los aminoácidos, flexibilidad, estructura secundaria, y dispersión de los ángulos diédricos. Encuentran que la precisión está influenciada por la flexibilidad pero no por la composición. Los residuos en hélices tienen mejor precisión comparado con las otras estructuras secundarias, aunque en algunas hojas betas también tienen alta precisión. La glicina tiene la peor precisión comparada con los otros aminoácidos. Su flexibilidad es mayor, reflejada en la gran dispersión de sus ángulos diédricos, contrario a los otros aminoácidos la precisión de ψ es mayor que φ. Explicando porque es difícil predecir los ángulos para este aminoácido y otros, como la asparagina que tiene propiedades similares.

En [5] se utiliza un Semi-Markov Conditional Random Fields para la predicción de genes *ab initio*, describiendo un programa llamado CRAIG. La predicción *ab initio* recae solo en el uso de características de los genes. Contrario a CRAIG, la mayoría de los predictores anteriores se basaron en el modelo generativo HMM para combinar contenido de la secuencia y clasificadores de señales en una consistente estructura de los genes. Sin embargo, la combinación del contenido de la secuencia, los clasificadores de señales y el modelo HMM de la estructura de los genes; no se entrena para maximizar la precisión de la predicción.

Del aspecto anterior, y basados en los buenos resultados de predicción de estructura de modelos que globalmente involucran la optimización de sus parámetros como los CRFs; proponen el uso de un Semi-CRF, con el fin de combinar muchas características diversas para lograr alta precisión en la predicción de los genes. Un Semi-CRF es un modelo que retorna la segmentación de una cadena de entrada con sus correspondientes etiquetas. Por la flexibilidad de los CRFs, usaron valiosas características que antes eran difíciles de integrar en modelos previos, por ejemplo, diferentes tipos de intrones de acuerdo a su tamaño. Esto permitió sobrepasar los resultados de los mejores predictores a la fecha en datos de prueba conocidos. Específicamente comparan CRAIG con GenScan, TwinScan 2.03, Genezilla, y Augustus. Sus avances mejoraron la sensibilidad y especificidad de predicciones de exones iniciales y solitarios en una media relativa de 25.5 % y 19.6 % respectivamente; al nivel de gen, la mejora de media relativa fue de 33.9 %; y la mejora de valor-F en la región codificante fue de 16.05 % a nivel de exon.

2.2.4.1. Predicción de pliegues con un CRF

Un paso para la predicción de la estructura 3D de una proteína, es identificar como varias estructuras secundarias adyacentes se organizan en el espacio; llamadas estructura supersecundaria o pliegues. El problema de predicción de pliegues es el siguiente: dada la secuencia de una proteína y un pliegue particular, predecir si la proteína adopta el pliegue estructural, y si lo hace, localizar las posiciones exactas de cada componente en la secuencia. Las herramientas de predicción de pliegues que se basan en perfiles y homólogos, funcionan bien para pliegues cortos con mucha similitud en secuencia. Sin embargo, existen pliegues con poca similitud e involucran interacciones a largo rango. Estos casos, necesitan un modelo más robusto, que capture las características de los pliegues sin necesidad de similitud en la secuencia. Es así como Liu *et al.* en [45], propusieron los *segmentation conditional random fields* (SCRFs) para predecir pliegues. Este modelo tiene todas las ventajas de los CRFs, y al mismo tiempo puede manejar observaciones de diferente tamaño. Además, la configuración del modelo gráfico es similar a la estructura 3D de las proteínas, y proporciona un *framework* para modelar interacciones largas entre estructuras secundarias. El modelo se aplicó al problema de predecir el pliegue beta-helix, con el resultado que sobrepasa a BetaWrap, un algoritmo para predecir pliegues beta-helix, y a HMMER una herramienta genérica de predicción de pliegues. Además, identificó proteínas en la base de datos Uniprot con pliegues beta-helix previamente desconocidas [45].

2.2.5. Software para modelos gráficos

A continuación se lista algunos productos de software disponibles que implementan modelos gráficos:

- MALLET - http://mallet.cs.umass.edu/
  Es un paquete implementado en Java con licencia CPL para aplicaciones de aprendizaje de maquina en texto, como: clasificación de documentos, agrupamiento, modelado de temas, y extracción de información. Para trabajar con secuencias el paquete proporciona implementaciones de modelos ocultos de Markov (HMM), modelos de

Markov de máxima entropía (MEMM), y campos aleatorios condicionales lineales (cadena lineal CRF). Contiene un paquete llamado GRMM que es una herramienta para desarrollar inferencia y aprendizaje en modelos gráficos lineales.

- FACTORIE - http://factorie.cs.umass.edu
  Es una herramienta implementada en Scala con licencia Apache versión 2.0 que proporciona un *framework* para trabajar con modelos gráficos. FACTORIE se describe como la herramienta más genérica que remplaza a MALLET en todo sentido. Pueden crearse grafos de factores utilizando programación declarativa e imperativa, realizar estimación de parámetros e inferencia. La herramienta tiene limitaciones para trabajar con variables continuas, porque esta implementada para trabajar con variables discretas. Tiene muchos modelos pre-construidos como regresiones lineales, Bayesiano ingenuo, SVM, árboles de decisión, cadenas lineales CRFs, modelamiento de temas, y herramientas para procesamiento del lenguaje natural.

- CRF++ - https://taku910.github.io/crfpp/
  Implementado en C++ con licencia GNU Lesser General Public o new BSD, es una herramienta que sirve para etiquetar datos secuenciales, implementa una cadena lineal CRF y puede ser utilizado en una gran cantidad de tareas de procesamiento de lenguaje natural como: reconocimiento de entidades y extracción de información. La herramienta tiene un formato para indicar la plantilla de las funciones características a utilizar en el CRF lineal, y otro formato para los archivos de entrenamiento y prueba. CRF++ automáticamente crea las funciones características de los archivos de entrenamiento y prueba a partir de la plantilla.

- CRFSuite - http://www.chokkan.org/software/crfsuite/
  Implementado en C++ con licencia BSD, es una herramienta que implementa una cadena lineal CRF de forma eficiente. Utiliza un formato simple para los datos de entrenamiento y prueba, que permite tener diferente número de características por etiqueta. Lo que hace que esta herramienta sea mas flexible en las funciones características que CRF++. Implementa diferentes métodos de entrenamiento y una utilidad para evaluar modelos entrenados.

- CRFSharp - https://github.com/zhongkaifu/CRFSharp
  Implementado en .NET(C#), es una herramienta que implementa una cadena lineal CRF con un eficiente uso de recursos computacionales. Utiliza los mismos formatos que CRF++ para definir la plantilla de funciones características y archivos de entrenamiento y prueba.

Capítulo 3

Predicción de alfabetos estructurales usando una cadena lineal CRF

Uno de los problemas más importantes en proteómica es predecir representaciones estructurales de las proteínas dada la secuencia de los aminoácidos. Una de las representaciones estructurales más conocida es la estructura secundaria, que clasifica las estructuras recurrentes en las proteínas en tres estados generales: hélices, hojas, y giros. La predicción de estructura secundaria consiste en asignar a segmentos de aminoácidos sus correspondientes estados. Muchos métodos de aprendizaje de máquina han sido aplicados a este problema, los mejores resultados para tres estados tienen alrededor de 80 % de precisión Q_3. La representación de la estructura secundaria carece de la característica de describir de forma adecuada la geometría 3D de las proteínas, ya que dos proteínas pueden tener asignaciones similares de estructura secundaria pero diferir mucho en su estructura 3D. Para superar esta limitación, los estudios se centraron en crear una representación llamada alfabeto estructural. Un alfabeto estructural es un conjunto mínimo de estructuras 3D recurrentes de las proteínas que describen la mayoría o todas las conformaciones del *backbone* de forma precisa [32], esta descripción estructural discretiza el espacio conformacional de las estructuras. Dependiendo de la técnica usada para hallar las estructuras, los alfabetos se caracterizan por la cantidad y tamaño de las estructuras. Los alfabetos son construidos pensando en que sean adecuados para describir el *backbone*; pueden ser buenos o no para utilizarlos en predicción de estructura.

Al igual que la predicción de estructura secundaria, la predicción de la estructura local de una proteína mediante un alfabeto estructural, consiste en asignar a segmentos de aminoácidos su elemento estructural correspondiente dada la información de la secuencia. Uno de los alfabetos estructurales más utilizados en predicción es PB *(protein blocks)*, propuesto por De Brevern et al. [19]. Está compuesto de 16 elementos estructurales de tamaño fijo, generados por un mapa organizacional mediante una medida de similitud de ángulos diédricos. Ha sido usado entre otros para: predecir el *backbone*, describir fragmentos estructurales largos o cortos, análisis de sitios de enlace, y alineaciones estructurales [30].

Aunque la estructura secundaria y los alfabetos estructurales son diferentes representaciones, éstos están relacionados. Los elementos estructurales de un alfabeto tienen asociadas preferencias a estructuras secundarias, y pueden utilizarse para analizar las estructuras, como en [25], donde usaron el alfabeto PB para analizar giros. En [16] se realizó un análisis de la correspondencia entre estructura secundaria y los elementos estructurales del alfabeto PB, para diferentes algoritmos de asignación de estructura secundaria. Analizaron la frecuencia de cada elemento estructural en los estados de estructura secundaria y se concluyó que los alfabetos estructurales son una herramienta adecuada para analizar las estructuras de las proteínas, se resaltaron las diferencias entre los resultados de las asignaciones de los diferentes algoritmos de asignación de estructura secundaria. Es importante destacar, que se observan patrones de frecuencias de ciertos elementos estructurales por determinadas estructuras secundarias, en donde se diferencian los elementos más frecuentes para hélices y hojas, pero se observan elementos que tienen frecuencias parecidas entre hélices y giros y entre hojas y giros. La frecuencia de los elementos entre hojas y giros es más parecida, lo que puede dificultar la discriminación entre los elementos frecuentes en los dos tipos de estructuras.

La predicción de elementos estructurales usando el alfabeto PB se mejoró con base en los avances de predicción de estructura secundaria, como es el caso de [56, 75, 23]. En [23] con la intuición que ambas representaciones tienen características comunes, ya que se derivan del *backbone* de la estructura de las proteínas y se evidencia que los elementos estructurales vecinos están correlacionados, como es el caso de las estructuras secundarias vecinas. De lo anterior, toman la idea del modelo de predicción de estructura secundaria de dos capas, que en su primera capa relaciona la secuencia-estructura, y en la segunda estructura-estructura. Además, los predictores toman como entrada la matriz de valoración específica de la posición (PSSM), ya que esta contiene información evolutiva y mejora significativamente la predicción de estructura secundaria [29]. Algunos predictores de alfabetos estructurales mejoraron sus resultados tomando como entrada la estructura secundaria predicha para explotar la correlación existente con los elementos estructurales [69][56].

De los avances en predicción de elementos estructurales, los aspectos más importantes a tener en cuenta para predecir son: primero, incorporar características de las proteínas que permitan discriminar localmente las preferencias secuencia-estructura de cada uno de los elementos estructurales, como por ejemplo: información evolutiva PSSM, información de los aminoácidos e información de la estructura secundaria; segundo, utilizar información de interacciones a largo rango entre los aminoácidos como por ejemplo, la utilización de dos clasificadores, el primero para relacionar secuencia-estructura y el segundo relaciona estructura-estructura. De hecho, el segundo aspecto es el más difícil de integrar y resolver en los modelos del estado del arte debido a limitaciones del modelo o recursos de computación. Por ejemplo, una de las interacciones a largo rango de aminoácidos son los enlaces de hidrógeno entre hojas beta, cuyos aminoácidos pueden estar muy separados en la cadena.

En otras palabras, en la predicción de estructura de proteínas se necesitan modelos que puedan integrar información de un contexto local de los aminoácidos, y también la información a largo rango. Pero como es descrito en [2], se presentan dos retos: primero, evitar el sobreajuste con largas ventanas de contexto; segundo, capturar la poca y débil información de largo rango necesaria para la predicción local, mientras se desecha el ruido

de la información de largo rango.

A continuación y en el capítulo siguiente se utilizan campos aleatorios condicionales CRFs para predecir la estructura local de las proteínas con dos alfabetos estructurales (PB y $SA_{10,3}$), con base en la hipótesis que los modelos gráficos son buenos para integrar la información de los aminoácidos y explotar la correlación entre las estructuras locales vecinas. Se evalúa la capacidad de diferentes funciones características de los aminoácidos para mejorar la predicción de estructura local. También se realiza un análisis de la relación entre los estados de la estructura secundaria definida por DSSP y los elementos de los alfabetos considerados.

3.1. Especificaciones de la cadena lineal CRF

La cadena lineal CRF se aplica al problema de predicción de alfabetos estructurales de la siguiente manera: Dado que el modelo gráfico codifica la distribución de probabilidad $p(Y \mid X)$, en donde los vértices del modelo son variables aleatorias, y la conexión entre los vértices representa una relación entre estas variables. Entonces, la secuencia de observaciones X corresponde a los aminoácidos $X = \{x_1, x_2, \ldots, x_n\}$, donde n es el tamaño de la proteína. La secuencia de etiquetas Y son los elementos estructurales de un alfabeto y corresponde a los elementos que codifican la proteína $Y = \{y_1, y_2, \ldots, y_m\}$, m es el número de elementos estructurales necesarios para codificarla, y $y_i \epsilon \{1, 2, \ldots, k\}$ donde k el número total de elementos estructurales en el alfabeto.

El modelo explotará la correlación existente entre los elementos estructurales vecinos, para hallar las relaciones estructura-estructura con la relación entre las etiquetas vecinas y_{t-1}, y_t. La información de los aminoácidos se utiliza para hallar las relaciones secuencia-estructura por medio de las funciones características. Es importante resaltar que la cadena lineal CRF integra la relación secuencia-estructura y estructura-estructura en un *framework* probabilístico.

3.1.1. Entrenamiento y predicción

El entrenamiento de la cadena lineal CRF se lleva a cabo seleccionando los parámetros que maximizan la probabilidad del conjunto de entrenamiento. En este trabajo se realiza, maximizando la log verosimilitud con la versión de memoria limitada de BFGS (algoritmo de Broyden–Fletcher–Goldfarb–Shanno). Esta técnica necesita calcular la log verosimilitud y sus derivadas, por lo que necesita hacer inferencia. La inferencia se realiza con las recursiones *forward* y *backward*.

Después que el modelo es entrenado, la predicción de las etiquetas para una observación se puede hallar con el algoritmo de Viterbi. Pero experimentalmente se han tenido mejores resultados calculando la probabilidad marginal para cada etiqueta, $p(y_i|x)$. Entonces la etiqueta con la probabilidad marginal más alta puede ser usada como la etiqueta predicha. Los algoritmos *forward-backward* y la estimación de parámetros se describen en la sección 2.2.3.4 del Capítulo 2.

3.1.2. Funciones características

La definición de cada función característica es el medio para lograr que el CRF logre modelar un sistema deseado. Las funciones características son dependientes del contexto del sistema a modelar. En el contexto de la predicción de elementos estructurales con una cadena lineal CRF, las funciones características son funciones de los elementos estructurales (las etiquetas) y de los aminoácidos (las observaciones). A continuación se describen las funciones características utilizadas en los experimentos:

1. **Observación de un aminoácido:** Esta característica captura la preferencia de los aminoácidos a estar relacionada con un elemento estructural por posición i en una ventana de observaciones.

2. **Observación de aminoácidos vecinos:** El objetivo de esta característica es capturar información de segmentos de aminoácidos por posición en la ventana de observaciones. Se entienden como las preferencias entre aminoácidos vecinos. Las características se conforman deslizando una ventana pequeña en la ventana de observaciones. Por ejemplo, si se tiene una ventana de cinco aminoácidos $\langle x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}\rangle$ siendo x_i el aminoácido central, y se desliza una ventana de tamaño dos, se obtendrán cuatro funciones características para las observaciones.

$$\langle x_{i-2}, x_{i-1}\rangle, \langle x_{i-1}, x_i\rangle, \langle x_i, x_{i+1}\rangle, \langle x_{i+1}, x_{i+2}\rangle$$

3. **Etiqueta de la predicción de la estructura secundaria:** La predicción de tres estados de la estructura secundaria de una proteína es muy valiosa, ya que la estructura secundaria está correlacionada con los elementos estructurales del alfabeto, y aprovecharla puede ayudar a discriminar entre los diferentes elementos. Se utiliza el programa PSSPRED [68] para predecir la estructura secundaria de todas las proteínas del conjunto de datos. Esta función característica captura la preferencia de cada estructura secundaria a estar relacionada con un elemento estructural en una ventana de observaciones.

4. **Etiquetas vecinas de la predicción de la estructura secundaria:** Esta función característica captura la relación entre vecinos de las estructuras secundarias en una ventana con los elementos estructurales del alfabeto. Las etiquetas vecinas se forman igual que la función de activación de la observación de aminoácidos vecinos, deslizando una ventana más pequeña en la ventana de etiquetas predichas.

5. **Puntajes de la predicción de las estructuras secundarias:** Para cada aminoácido de una proteína se tendrán las funciones características que corresponden a los puntajes arrojados por el predictor PSSPRED de estructura secundaria. Para cada elemento estructural e del alfabeto y estado es de la estructura secundaria, se tienen funciones de activación de la siguiente manera:

$$f_{e,es}(x, y_t) = \begin{cases} g_{es}(x_t) & si\, y_t = e \\ 0 & de\, otra\, forma \end{cases}$$

   Donde $g_{es}(x_t)$ retorna la probabilidad predicha por el predictor de estructura secundaria para el estado es del aminoácido x_t.

6. **Asignación de la estructura secundaria:** El programa utilizado para la asignación de estructura secundaria es DSSP [33], está basado en patrones de enlaces de hidrógeno y define ocho estados. Los ocho estados son agrupados para obtener tres de la siguiente manera: las hélices se asocian con los estados α-helix, 3_π-helix, π-helix; las hojas con β-strand y β-sheet; y los giros con turn, bend, y coil. Se utiliza la información de la asignación de la estructura secundaria como una función característica para determinar los límites superiores en la exactitud de la predicción.

7. **Matriz PSSM:** La matriz PSSM (*position-specific scoring matrix*) que contiene información evolutiva, se deriva de secuencias de homólogos y es dependiente de la posición en la secuencia. Es de tamaño $n \times 20$, donde n es el tamaño de la proteína y proporciona un puntaje de sustitución para cada uno de los aminoácidos en cada posición. Se obtiene por medio del programa de alineación PSI-BLAST con cinco iteraciones y E-value = 0.001, utilizando una base de datos no redundante NR. Previamente la base de datos fue filtrada, removiendo secuencias con baja información y regiones de hélice superenrollada con el programa pfilt del paquete PSIPRED [29]. La función característica es descrita a continuación:

$$f_e\left(x, y_t\right) = \begin{cases} t\left(x_t\right) & si\, y_t = e \\ 0 & de\, otra\, forma \end{cases}$$

   Para cada elemento estructural e del alfabeto, $t\left(x_t\right)$ retorna el puntaje de sustitución para el aminoácido x_t transformado linealmente al intervalo $[0, 1]$ de acuerdo a Kim et al [35], para tener buen rendimiento:

$$t\left(x\right) = \begin{cases} 0 & si\, x \leq -5 \\ \frac{1}{2} + \frac{x}{10} & si\ -5 < x < 5 \\ 1 & si\, x \geq 5 \end{cases}$$

8. **Características físico-químicas de los aminoácidos:** Se usa la información físico-química de los aminoácidos definida por [50], que define para cada aminoácido los valores de parámetro estérico, polarizabilidad, volumen, hidrofobicidad, y punto isoeléctrico.

3.2. Materiales y experimentos

En este trabajo se uso un proceso empírico para distinguir las funciones características que mejoran la precisión del modelo. Por lo anterior, combinaciones de funciones características son utilizadas en experimentos para evaluar su capacidad de mejoramiento en la precisión del modelo. Los resultados de los experimentos se obtienen por validación cruzada de cinco iteraciones sobre el conjunto de datos; 4/5 de los datos se utilizan para entrenar y 1/5 para de los datos para validar.

3.2.1. Conjuntos de datos

Se utiliza un subconjunto no redundante de la base de datos PDB. El conjunto tiene 3052 secuencias y estructuras de proteínas no homologas (después de todos los filtros), seleccionadas con el servidor web PISCES Cull PDB [63], que es utilizado comúnmente para seleccionar conjunto de datos para predicción de estructura. Las proteínas son extraídas con los siguientes criterios: identidad en la secuencia entre cualquier par de proteínas menor que 25 %, solo estructuras resueltas por rayos-X, proteínas con estructuras resueltas con resolución mejor que 2.5 Å, R-factor mejor que 0.2. Se excluye las proteínas con menos de 50 o más de 800 aminoácidos o cadenas discontinuas.

3.2.2. Alfabetos estructurales

Se utiliza el alfabeto estructural PB descrito en De Brevern *et al.* [19]. Contiene 16 elementos estructurales asignados con las letras de a hasta p. Fue construido usando un mapa organizacional, con fragmentos de 5 aminoácidos, descritos con 8 ángulos diédricos asociados a los 5 carbonos alfa. La medida de similitud utilizada en el proceso de agrupamiento es raíz de la media de los cuadrados de los errores de valores angulares RMSDA. Además tuvo en cuenta las transiciones entre los elementos estructurales en el proceso de agrupamiento, con el fin de mejorar las características de predicción. Este alfabeto ha sido usado para describir el *backbone* de las proteínas y para realizar predicciones locales de estructura. Sus elementos estructurales tienen relación con las estructuras secundarias, el elemento estructural m es el patrón principal de las hélices alfa, los elementos k, l y n, o, p son referentes a hélice n-cap y c-cap respectivamente. El elemento d corresponde a los patrones de hojas beta, los elementos $a - c$ y $d - f$ tienden a encontrarse en hojas beta. Los elementos de g hasta j, principalmente se encuentran en giros.

En este trabajo, para codificar la estructura de una proteína por medio de los elementos estructurales, se utiliza la codificación local. Para codificar una proteína con el alfabeto PB, se usa una venta deslizante de 5 aminoácidos, el residuo central es asignado con el elemento estructural que tenga menor RMSDA. Así, una proteína de tamaño n, se describe por $n - 4$ elementos estructurales. En [17] se mostró que el alfabeto PB tiene una aproximación estructural RMSD (distancia Euclidiana promedio entre Cα sobrepuestos) media de 0.41 Å con una desviación estándar de 0.25 Å y mediana de 0.34 Å. Para cada PB, el RMSD medio se halla sobreponiendo todos los fragmentos asociados a un PB y el prototipo 3D del PB.

También, se utiliza el alfabeto $SA_{10,3}$ [62] que está formado por 10 elementos estructurales asignados con letras a hasta j y fragmentos de tres aminoácidos. Los fragmentos se representan por las coordenadas de los átomos de nitrógeno N, carbono alfa $C\alpha$, y carbono C. Una característica importante de esta representación es que permite una reescritura directa de los símbolos estructurales a las coordenadas 3D sin pérdida de información estructural del *backbone*, ya que no hay transformación de la información. Al contrario, el alfabeto PB tiene que calcular las coordenadas de los carbono alfa a partir de los ángulos diédricos, lo que significa que en la reescritura pierde información estructural. El alfabeto $SA_{10,3}$ fue construido con una extensión del algoritmo k-means, que usa simulated annealing para

encontrar un agrupamiento óptimo basado en el criterio de Silhouette. La medida de similitud utilizada en el proceso de agrupamiento es raíz de la media de los cuadrados de los errores RMSD. Para la información detallada del algoritmo de agrupamiento remitirse a [62].

Como se mencionó anteriormente, para codificar una proteína con un alfabeto estructural se utiliza la codificación local. Es así, que para $SA_{10,3}$ se desliza una ventana de 3 aminoácidos en la secuencia de la proteína, el residuo central es asignado con el elemento estructural que tenga menor raíz de la media de los cuadrados de los errores RMSD. Entonces, una proteína de tamaño n, se describe por $n - 3$ elementos estructurales. En un conjunto independiente al de entrenamiento, el alfabeto $SA_{10,3}$ tiene un error 0.41 ± 0.18 de RMSD en la codificación local. Los alfabetos estudiados en [62] se caracterizan por describir bien el *backbone* de la proteína con menor número de elementos estructurales, comparado con otros estudios.

3.2.3. Evaluación de los resultados del modelo

La calidad de los modelos se evalúa con la exactitud, precisión, sensibilidad, y coeficiente de correlación de Matthews MCC. Todas estas métricas usan las definiciones de la tabla 3.1 para problemas de clasificación binaria supervisada (verdaderos positivos tp, falsos positivos fp, verdaderos negativos tn, y los falsos negativos fn).

		Predicción	
		Verdadero	**Falso**
Observación	**Verdadero**	tp	fn
	Falso	fp	tn

Tabla 3.1: Matriz de confusión de clasificación binaria.

La exactitud Q_k es usada para evaluar la predicción del modelo, esta se refiere al porcentaje de elementos estructurales correctamente predichos. La exactitud Q_k para una predicción de k clases se define como:

$$Q_k = \frac{1}{N} \sum_{i=1}^{k} tp_i$$

Donde N es el número total de elementos estructurales, y tp_i es la cantidad de verdaderos positivos para la clase i. Se conoce que la precisión no es una métrica fiable para datos desequilibrados (como el caso de la predicción de elementos estructurales), pero será dada para permitir comparaciones del conjunto de datos.

Además, se usa la precisión que se define como la fracción de los elementos predichos correctos, y la sensibilidad como la fracción de los elementos observados predichos correctamente:

$$Precisión = \frac{tp}{tp + fp}$$

$$Sensibilidad = \frac{tp}{tp + fn}$$

También se usa el coeficiente de correlación de Matthews MCC, por su utilidad para evaluar conjunto de datos desequilibrados. Es una medida de correlación entre las observaciones y las clasificaciones predichas, sus valores están entre 1 y -1. Un coeficiente de 1 representa una predicción perfecta, 0 no mejor que una predicción aleatoria, y -1 total desacuerdo entre las observaciones y las predicciones:

$$MCC = \frac{tp \times tn - fp \times fn}{\sqrt{(tp + fp)(tp + fn)(tn + fp)(tn + fn)}}$$

3.3. Resultados y análisis

3.3.1. Análisis de la relación entre la estructura secundaria y los alfabetos estructurales PB y $SA_{10,3}$

Se realiza un análisis de la frecuencia entre la asignación de los elementos estructurales y los estados de estructura secundaria asignados por DSSP con el fin de caracterizar las preferencias de los elementos estructurales y los estados de la estructura secundaria en el conjunto de datos de predicción. En [16] se realiza un análisis extenso de correspondencia entre el alfabeto PB y diferentes algoritmos de asignación de estructura secundaria.

Los alfabetos estructurales PB - $SA_{10,3}$ y la relación con la asignación de tres estados de estructura secundaria con DSSP

Las tablas 3.2 y 3.3 muestran las frecuencias de los alfabetos PB y $SA_{10,3}$ en el conjunto de datos, y la relación con la asignación de estructura secundaria con la asignación de tres estados según DSSP. Los ocho estados que considera DSSP son agrupados para obtener tres de la siguiente manera: las hélices se asocian con los estados α-helix, 3_π-helix, π-helix; las hojas con β-strand y β-sheet; y los giros con turn, bend, y coil. La frecuencia de los elementos de ambos alfabetos estructurales en el conjunto de datos esta desequilibrado, los elementos más frecuentes en ambos alfabetos corresponden a los asociados con las hélices y hojas.

La tabla 3.2 muestra las frecuencias para el alfabeto PB. Los elementos estructurales más frecuentes son m y d correspondientes a hélices y hojas, respectivamente. Los elementos n y l se asocian con mayor frecuencia a hélices. El elemento k aparece en hélices y giros con un frecuencia similar, los elementos o y p se asocian a hélices, pero su frecuencia es mayor en giros. Los elementos e y c son frecuentes en hojas, pero c es más frecuente en giros. Los elementos a, b, c, e, f, g, h, i, j, o, y p son frecuentes en giros, pero se encuentran

con menor frecuencia en hojas y hélices, el elemento g es el que tiene mayor frecuencia en los tres estados. La mayoría de elementos asociados a giros se encuentran en hojas, lo que puede indicar una dificulta en los predictores para discriminar elementos estructurales asociados a hojas y giros. Las frecuencias de los elementos en el conjunto de datos son similares a las frecuencias del trabajo [16].

La tabla 3.3 muestra las frecuencias para el alfabeto $SA_{10,3}$. Los elementos más frecuentes son b e i correspondientes a hélices y hojas, respectivamente. Los elementos a, c, d, y f se relacionan con hélices pero son muy frecuentes en giros. El elemento j es muy frecuente en hojas. Los elementos g y h son muy frecuentes en giros pero también se encuentran en hojas. El elemento e es exclusivo en giros. Se observa que los elementos asociados con hélices son encontrados con mayor frecuencia en giros, a diferencia del elemento b. Además, los elementos asociados con hojas son frecuentes en giros, a diferencia del elemento j. Lo anterior hace probable que un predictor tenga dificulta al discriminar entre hélices-giros y hojas-giros.

PB	**Frecuencia relativa**	**Hélices**	**Hojas**	**Giros**
a	3.82	0.02	21.19	78.78
b	4.45	0.22	13.74	86.04
c	8.48	0.80	45.56	53.64
d	18.66	0.00	74.25	25.75
e	2.21	0.17	55.61	44.22
f	6.62	0.00	28.85	71.15
g	1.13	19.21	10.43	70.36
h	2.19	4.27	21.41	74.32
i	1.54	2.56	6.58	90.86
j	0.88	9.11	11.79	79.10
k	5.45	49.66	0.56	49.78
l	5.22	64.75	0.84	34.41
m	31.63	90.94	0.20	8.86
n	1.81	75.09	0.45	24.45
o	2.52	29.18	0.57	70.25
p	3.39	20.19	0.94	78.87

Tabla 3.2: Frecuencias en tres estados de la asignación de estructura secundaria según DSSP para cada elemento del alfabeto PB en el conjunto de datos.

$SA_{10,3}$	**Frecuencia relativa**	**Hélices**	**Hojas**	**Giros**
a	5.11	44.19	1.23	54.58
b	31.33	95.69	0.03	4.28
c	3.69	47.22	0.60	52.19
d	10.40	29.29	6.95	63.76
e	4.25	1.95	2.92	95.13
f	3.38	22.99	1.43	75.58
g	10.07	0.02	26.47	73.51
h	7.68	0.48	24.43	75.09
i	14.11	0.00	70.22	29.78
j	9.98	0.00	77.86	22.14

Tabla 3.3: Frecuencias en tres estados de la asignación de estructura secundaria según DSSP para cada elemento del alfabeto $SA_{10,3}$ en el conjunto de datos.

Los alfabetos estructurales PB-$SA_{10,3}$ y la relación con la asignación de estructura secundaria con DSSP

Las tablas 3.4 y 3.5 muestran las frecuencias de los alfabetos PB y $SA_{10,3}$ en el conjunto de datos y la relación con la asignación DSSP de estructura secundaria.

La tabla 3.4 muestra las frecuencias de los elementos del alfabeto PB. Los elementos con mayor frecuencia en el estado α-helix son m (86 %), n (67 %), l (46 %), k (36 %), y o (23 %). En el estado 3_π-helix los más frecuentes son l (18 %), k (13 %), p (12 %), y g (12 %). El estado π-helix no es frecuente y las frecuencias de los elementos son bajas. Los elementos más frecuentes en el estado β-strand son d (72 %), e (52 %), c (41 %), f (25 %), h (19 %), y a (18 %). En el estado β-sheet las frecuencias de los elementos son bajas y no tiene elementos representativos. Los elementos más frecuentes para el estado Turn son i (67 %), o (60 %), h (48 %), p (47 %), k (34 %), j (26 %), l (22 %), n (19 %), g (18 %), y b (16 %). Los elementos más frecuentes en el estado Bend son b (44 %), j (31 %), g (25 %), i (21 %), a (16 %), c (14 %), y p (14 %). Los elementos más frecuentes en el estado Coil son f (62 %), a (58 %), c (38 %), e (36 %), g (25 %), b (24 %), y j (21 %).

Se observa que los elementos frecuentes en las hélices también tienen frecuencias altas en el estado Turn, aun con esta observación se pueden definir elementos exclusivos de las hélices. Las hojas aunque tienen elementos frecuentes, estos elementos también son frecuentes en el estado Coil y no tienen exclusividad en los estados hojas, a diferencia del elemento d que es exclusivo. Las frecuencias de los elementos en el conjunto de datos son similares a las frecuencias del trabajo [16].

La tabla 3.5 muestra las frecuencias de los elementos del alfabeto $SA_{10,3}$. Los elementos con mayor frecuencia en el estado α-helix son b (90 %), c (40 %), a (24 %), d (20 %), y f (17). En el estado 3_π-helix el frecuente es a (20 %). El estado π-helix no es frecuente y las frecuencias de los elementos son bajas, pero el elemento c (0.3 %) es el más frecuente. Los elementos más frecuentes en el estado β-strand son j (75 %), i (68 %), g (22 %), h (20 %). En el estado β-sheet las frecuencias de los elementos son bajas y no tiene elementos representativos. Los elementos más frecuentes para el estado Turn son e (56 %), f (50 %), c

(35 %), *a* (34 %), y *d* (27 %). Los elementos más frecuentes en el estado Bend son *g* (30 %), *d* (24 %), *e* (21 %), *f* (16 %), y *c* (15 %). Los elementos más frecuentes en el estado Coil son *h* (66 %), *g* (42 %), *i* (29 %), *j* (22 %), *e* (17 %), *a* (14 %). El elemento *e* es exclusivo de giros con una frecuencia mayor en el estado Turn. Las hojas tienen exclusividad con los elementos *j* e *i*. El elemento *b* es exclusivo de las hélices. Los elementos asociados a hélices también son frecuentes en el estado Turn y los elementos frecuentes en hojas son frecuentes en el estado Giros.

	Hélices			**Hojas**		**Giros**		
PB	**α-helix**	**3_π-helix**	**π-helix**	**β-sheet**	**β-strand**	**Coil**	**Bend**	**Turn**
a	0.00	0.02	0.00	2.77	18.42	58.70	16.66	3.42
b	0.06	0.15	0.01	0.07	13.67	24.38	44.77	16.89
c	0.08	0.72	0.00	3.95	41.61	38.28	14.50	0.86
d	0.00	0.00	0.00	1.57	72.68	20.54	5.12	0.09
e	0.01	0.15	0.01	2.81	52.80	36.36	6.98	0.89
f	0.00	0.00	0.00	3.71	25.14	62.84	7.69	0.62
g	6.70	12.49	0.01	2.65	7.78	25.69	25.94	18.74
h	0.23	4.02	0.01	2.01	19.40	14.41	11.75	48.16
i	0.13	2.43	0.00	0.16	6.42	8.16	21.55	61.14
j	3.92	5.17	0.02	1.90	9.89	21.13	31.07	26.90
k	36.02	13.63	0.02	0.01	0.55	6.32	8.66	34.81
l	46.49	18.23	0.03	0.27	0.57	5.29	6.66	22.46
m	86.41	4.49	0.04	0.09	0.11	2.48	1.46	4.92
n	67.77	7.28	0.04	0.22	0.23	2.12	3.18	19.15
o	23.33	5.84	0.01	0.47	0.09	2.95	6.60	60.71
p	8.06	12.12	0.01	0.17	0.77	16.99	14.62	47.26

Tabla 3.4: Frecuencias en los ocho estados de la asignación de estructura secundaria según DSSP para cada elemento del alfabeto PB en el conjunto de datos.

	Hélices			**Hojas**		**Giros**		
$SA_{10,3}$	**α-helix**	**3_π-helix**	**π-helix**	**β-sheet**	**β-strand**	**Coil**	**Bend**	**Turn**
a	24.04	20.13	0.02	0.37	0.86	14.30	6.03	34.25
b	90.27	5.41	0.01	0.00	0.03	0.06	0.25	3.97
c	40.06	6.84	0.32	0.20	0.40	0.95	15.52	35.72
d	20.64	8.64	0.01	0.37	6.58	11.56	24.48	27.72
e	0.20	1.74	0.01	0.17	2.75	17.21	21.30	56.62
f	17.12	5.86	0.01	0.55	0.88	8.35	16.88	50.35
g	0.01	0.02	0.00	3.55	22.92	42.67	30.69	0.14
h	0.24	0.25	0.00	3.62	20.81	66.57	5.51	3.01
i	0.00	0.00	0.00	1.82	68.40	29.35	0.37	0.07
j	0.00	0.00	0.00	2.31	75.56	22.05	0.08	0.00

Tabla 3.5: Frecuencias en los ocho estados de la asignación de estructura secundaria según DSSP para cada elemento del alfabeto $SA_{10,3}$ en el conjunto de datos.

3.3.2. Evaluación de diferentes funciones características de los aminoácidos en la predicción de estructura local con alfabetos estructurales

Se llevan a cabo experimentos que combinan diferentes funciones características de los aminoácidos, con el fin de explotar la capacidad de los CRFs para integrar información diversa de las observaciones e identificar las características que mejoran la predicción de alfabetos estructurales. Los resultados de los experimentos son los promedios Q_k de la validación cruzada de cinco iteraciones en el conjunto de datos.

3.3.2.1. Selección del tamaño de ventana

La cadena lineal CRF se ve afectada por el factor de regularización para entrenar el modelo y la ventana de observaciones usada en las funciones características. El tamaño de la ventana de las observaciones es importante porque afecta la cantidad de funciones características del modelo y puede afectar la exactitud de la predicción. Por tal razón, se ejecutan experimentos variando el tamaño de la ventana para escoger la que obtenga buen rendimiento Q_k.

Los experimentos usan la misma configuración de funciones características, y son las funciones relacionadas con la información de los aminoácidos: observación de un aminoácido, observación de aminoácidos vecinos y perfil PSSM. Las tablas 3.6 y 3.7 contienen los resultados Q_k para tres diferentes tamaños de ventanas para el alfabeto PB y $SA_{10,3}$, respectivamente. El tamaño de ventana seleccionada para ambos alfabetos es de tamaño 5 con un Q_{16} de 57.44 % para PB y Q_{10} de 51.87 % para $SA_{10,3}$.

Tamaño Ventana	**Predicción Q_{16}**
5	57.44
7	57.29
9	56.73

Tabla 3.6: Rendimiento Q_{16} por tamaño de ventana para el alfabeto PB.

Tamaño Ventana	**Predicción Q_{10}**
3	49.52
5	51.87
7	52.11

Tabla 3.7: Rendimiento Q_{10} por tamaño de ventana para el alfabeto $SA_{10,3}$.

3.3.2.2. Experimentos con información de los aminoácidos y predicción de la estructura secundaria

Se conforman seis experimentos con información de los aminoácidos y predicción de la estructura secundaria. La tabla 3.8 muestra las combinaciones de las características utilizadas en cada experimento. La descripción de las características usadas, se encuentran en la sección 3.1.2.

El experimento base es $Exp1$, solamente usa las observaciones de los aminoácidos. Los experimentos $Exp2$ y $Exp3$ usan las características del experimento base y agregan la predicción de estructura secundaria; el experimento $Exp2$ usa las etiquetas predichas y

el experimento $Exp3$ los puntajes predichos. El experimento $Exp4$ usa las características del experimento base y agrega la información del perfil PSSM. Los experimentos $Exp5$ y $Exp6$ usan todas las características: las observaciones de los aminoácidos, la predicción de estructura secundaria, y el perfil PSSM; el experimento $Exp5$ utiliza las etiquetas predichas de la estructura secundaria y $Exp6$ los puntajes predichos.

La cadena lineal CRF usa la regularización L2 para prevenir el sobre entrenamiento, el factor de regularización es denotado por la variable λ. Determinar el mejor valor de regularización requiere computación intensiva, en este trabajo se evalúan cuatro valores de regularización $\lambda = \{1, 4, 9, 19\}$.

Los resultados Q_k de las predicciones con diferentes valores del factor de regularización se muestran en las tablas 3.9 y 3.10, y en la figura 3.1.

Características	$Exp1$	$Exp2$	$Exp3$	$Exp4$	$Exp5$	$Exp6$
Observación de un aminoácido y aminoácidos vecinos	+	+	+	+	+	+
Etiqueta y etiquetas vecinas de la predicción de la estructura secundaria	-	+	-	-	+	-
Puntajes de la predicción de las estructuras secundarias	-	-	+	-	-	+
Perfil PSSM	-	-	-	+	+	+

Tabla 3.8: Configuración de experimentos para evaluar información de los aminoácidos y predicción de la estructura secundaria.

Los mejores resultados para ambos alfabetos se obtienen con el factor de regularización $\lambda = 19$ o $\lambda = 9$. A medida que se aumenta el factor de regularización, los resultados del experimento $Exp1$ en ambos alfabetos empeoran. Y lo mismo sucede con los resultados del experimento $Exp4$ para el alfabeto $SA_{10,3}$.

A continuación se hace un análisis de los resultados con el factor de regularización $\lambda = 9$ por facilidad, pero el análisis es consistente con los diferentes valores de regularización. El experimento base $Exp1$ usa las observaciones de los aminoácidos, obtiene una exactitud Q_k de 42.28 % para PB y 37.31 % para $SA_{10,3}$. El experimento $Exp2$ agrega las etiquetas predichas de estructura secundaria al base, obtiene una exactitud de 59.71 % para PB y 53.95 % para $SA_{10,3}$. Tener en cuenta las etiquetas predichas aumenta la exactitud 17.43 % para PB y 16.64 % para $SA_{10,3}$. El experimento $Exp3$ agrega los puntajes predichos de estructura secundaria al base, obtiene una exactitud de 62.01 % para PB y 56.07 % para $SA_{10,3}$. Tener en cuenta los puntajes predichos aumenta la exactitud 19.82 % para PB y 18.76 % para $SA_{10,3}$. Se compara los resultados del experimento $Exp2$ y $Exp3$ para evaluar si es mejor usar las etiquetas o puntajes predichos de estructura secundaria, el experimento $Exp3$ tiene una mejora de 2.3 % para PB y 2.12 % para $SA_{10,3}$. Se concluye que usar la información de los puntajes predichos proporciona mejores resultados. La mejora en la predicción de alfabetos estructurales con el uso de la estructura secundaria predicha, concuerda con los resultados de los trabajos [24][69][56].

El experimento $Exp4$ agrega la matriz PSSM al experimento base, este obtiene 58.12 % para PB y 51.4 % para $SA_{10,3}$. Tener en cuenta la matriz PSSM aumenta la exactitud 15.84 % para PB y 14.09 % para $SA_{10,3}$. Es de resaltar que la información PSSM es la entrada por defecto en muchos predictores en una gran cantidad de problemas de predicción de estructura de proteínas, debido a que se obtienen muy buenos resultados. Se compara los resultados del experimento $Exp4$ y $Exp3$ para evaluar que característica entre la matriz

PSSM y los puntajes predichos de estructura secundaria proporcionan el mejor resultado. Los puntajes de estructura secundaria en el experimento $Exp3$ superan a la información de la matriz PSSM en el experimento $Exp4$ por 3.89 % para PB y 4.67 % para $SA_{10,3}$.

Los experimentos $Exp5$ y $Exp6$ usan las observaciones de los aminoácidos, matriz PSSM, y la estructura secundaria predicha. Se diferencian en que el experimento $Exp5$ usa las etiquetas predichas y $Exp6$ los puntajes predichos. El experimento $Exp5$ obtiene 63.48 % para PB y 57.88 % para $SA_{10,3}$. El experimento $Exp6$ obtiene 64.51 % para PB y 58.99 % para $SA_{10,3}$. Al igual que se mencionó anteriormente, la información de los puntajes de la predicción de la estructura secundaria en el experimento $Exp6$ proporcionan mejores resultados que usar las etiquetas predichas del experimento $Exp5$, con una ganancia de 1.03 % para PB y 1.11 % para $SA_{10,3}$. La combinación de las tres características en los experimentos $Exp5$ y $Exp6$ superan a los resultados de usarlas individualmente en los experimentos $Exp3$ y $Exp4$.

Experimentos	$\lambda = 1$	$\lambda = 4$	$\lambda = 9$	$\lambda = 19$
$Exp1$	43.03	42.76	42.28	41.62
$Exp2$	58.45	59.39	59.71	59.80
$Exp3$	60.86	61.77	62.01	62.08
$Exp4$	57.44	58.09	58.12	57.96
$Exp5$	62.12	63.13	63.48	63.63
$Exp6$	63.25	64.25	64.51	64.63

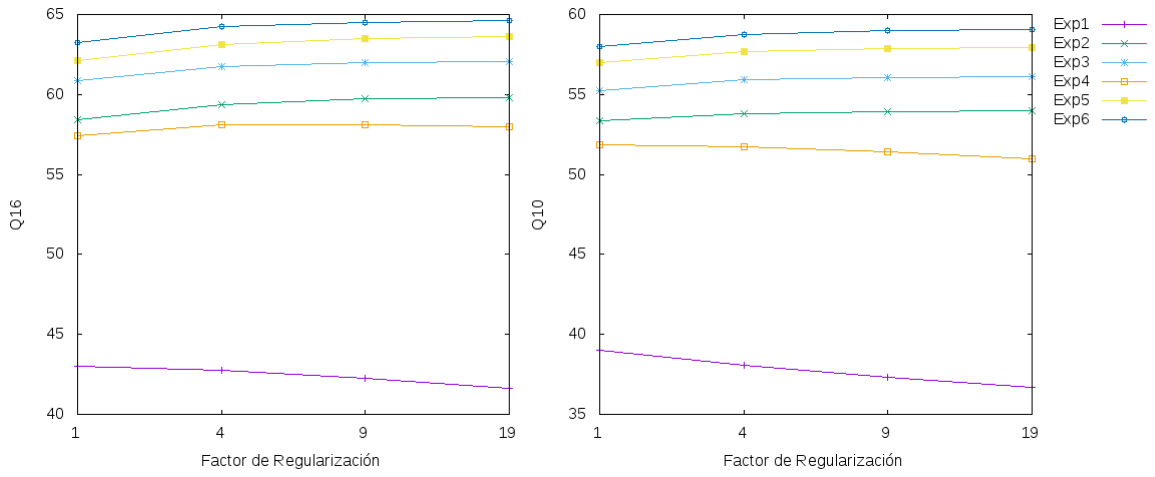
Tabla 3.9: Resultados de la exactitud Q_{16} del alfabeto PB para diferentes valores de regularización.

Experimentos	$\lambda = 1$	$\lambda = 4$	$\lambda = 9$	$\lambda = 19$
$Exp1$	39.02	38.05	37.31	36.71
$Exp2$	53.35	53.82	53.95	53.97
$Exp3$	55.21	55.91	56.07	56.11
$Exp4$	51.87	51.72	51.40	50.99
$Exp5$	56.97	57.66	57.88	57.91
$Exp6$	57.97	58.76	58.99	59.03

Tabla 3.10: Resultados de la exactitud Q_{10} del alfabeto $SA_{10,3}$ para diferentes valores de regularización.

El mejor resultado para ambos alfabetos estructurales se obtiene con el experimento $Exp6$ que utiliza la información de los aminoácidos, la matriz PSSM , y el puntaje de la predicción de la estructura secundaria. Los resultados son 64.63 % para PB y 59.03 % para $SA_{10,3}$ con $\lambda = 19$. El peor resultado para ambos alfabetos se obtiene con el experimento $Exp1$ que solo usa la información de los aminoácidos. Los resultados son 43.03 % para PB y 39.02 % para $SA_{10,3}$ con $\lambda = 1$.

Los resultados de los experimentos, muestran que el alfabeto PB es más predecible que $SA_{10,3}$, aunque tiene mayor número de elementos estructurales. En el contexto de los CRFs, se cree que la superioridad se debe en parte a que en el proceso de obtención de los PBs, se tiene en cuenta las transiciones entre los elementos estructurales, y la cadena



(a) (b)

Figura 3.1: Resultados de exactitud de los experimentos con diferentes valores de regularización. (a) Resultado Q_{16} para el alfabeto PB y (b) Resultado Q_{10} para el alfabeto $SA_{10,3}$.

lineal CRF explota las relaciones entre etiquetas vecinas. En el trabajo [34] el alfabeto PB mostró ser muy predecible comparado con otros alfabetos.

3.3.2.3. Experimentos con la asignación de estructura secundaria

En esta sección se hallan los límites superiores de predicción con la asignación de la estructura secundaria por medio de DSSP.

Se realizan dos experimentos $Exp2'$ y $Exp5'$, estos corresponden a los experimentos $Exp2$ y $Exp5$ pero en lugar de las etiquetas predichas usan las etiquetas asignadas. La tabla 3.11 muestra las configuraciones y resultados Q_k para los experimentos con factor de regularización $\lambda = 9$.

El experimento $Exp2'$ representa el límite superior del experimento $Exp2$, para el alfabeto PB es 66.64 % y para $SA_{10,3}$ es 60.38 %. El experimento $Exp5'$ es el límite superior del experimento $Exp5$, para el alfabeto PB es 70.58 % y para $SA_{10,3}$ es 64.25 %.

El resultado del experimento $Exp5$ está por debajo del límite superior por 7.1 % para PB y 6.37 % para $SA_{10,3}$. El resultado del experimento $Exp6$ que usa los puntajes de la estructura secundaria predicha está por debajo del límite superior por 6.07 % para PB y 5.26 %. Esto indica que la cadena lineal CRF aprende a corregir el error de la predicción de estructura secundaria para su beneficio. Este comportamiento se evidencia en el trabajo de [44], donde concluye que la cadena lineal CRF es un buen modelo de combinación de puntajes para predecir estructura secundaria.

Características	$Exp2'$	$Exp5'$
Observación de un aminoácido y aminoácidos vecinos	+	+
Perfil PSSM	-	+
Etiqueta y etiquetas vecinas de la asignación de estructura secundaria	+	+
Resultados	$Exp2'$	$Exp5'$
PB	66.67	70.58
$SA_{10,3}$	60.41	64.25

Tabla 3.11: Configuración y resultados Q_k de experimentos con la asignación de estructura secundaria para los alfabetos PB y $SA_{10,3}$.

3.3.2.4. Experimentos con la información físico-químicas de los aminoácidos

En esta sección se evalúa las características físico-químicas de los aminoácidos en la predicción de los alfabetos estructurales. Se utiliza las funciones características físico-químicas descrita en la sección 3.1.2. El mejor experimento de la sección 3.3.2.2 es seleccionado para agregarle las características físico-químicas. Se realiza el experimento $Exp6'$, que corresponde al experimento $Exp6$ agregando las características físico-químicas. La tabla 3.12 muestra la configuración y resultados Q_k del experimento con el factor de regularización $\lambda = 9$.

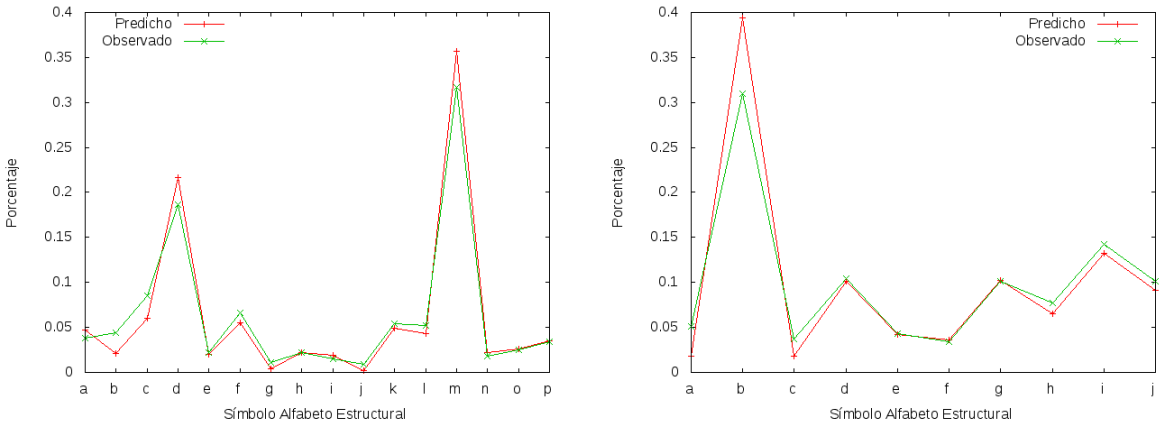
El resultado del experimento $Exp6'$ es 64.45 % para PB y 58.75 % para $SA_{10,3}$. Al comparar con el resultado del experimento $Exp6$, se disminuye la exactitud en 0.06 % para PB y 0.24 % para $SA_{10,3}$. Lo anterior muestra que las funciones características físico-químicas evaluadas no contribuyen a mejorar la exactitud de la predicción.

Características	$Exp6'$
Observación de un aminoácido y aminoácidos vecinos	+
Puntajes de la predicción de las estructuras secundarias	+
Perfil PSSM	+
Características físico-químicas	+
Resultados	$Exp6'$
PB	64.45
$SA_{10,3}$	58.75

Tabla 3.12: Configuración y resultados Q_k de experimentos con características físico-químicas para los alfabetos PB y $SA_{10,3}$.

3.3.3. La predicción está bien distribuida en los elementos estructurales

La figura 3.2 muestra el porcentaje de los diferentes elementos estructurales en las secuencias observadas y predichas para la validación cruzada del experimento $Exp6$ con factor de regularización $\lambda = 9$. En la gráfica los porcentajes de los elementos en las secuencias predichas y observadas son similares para ambos alfabetos estructurales. Además se evidencia que los elementos estructurales están desequilibrados.



(a) (b)

Figura 3.2: Porcentaje de elementos estructurales observados y predichos (a) para el alfabeto PB y (b) $SA_{10,3}$.

A continuación, se realiza un análisis teniendo en cuenta la relación entre elementos estructurales y estructura secundaria de la sección 3.3.1 para resaltar la predictibilidad de los elementos estructurales. Las tablas 3.13 y 3.14 muestran la precisión, sensibilidad, y MCC para cada elemento de los alfabetos estructurales PB y $SA_{10,3}$, respectivamente.

Para el alfabeto PB, los elementos m, n, o, k, l, y p asociados a hélices son más predecibles. El elemento m tiene la mejor predictibilidad y es muy frecuente, los elementos n, o, k, l, y p tienen buena predictibilidad a pesar de su poca frecuencia. Los elementos b, c, e, y h asociados a hojas y giros, tienen baja predictibilidad. A diferencia de los elementos anteriores, los elementos f y a que también son frecuentes en hojas y giros tienen predictibilidad regular. El elemento d tiene buena predictibilidad, es exclusivo de hojas y muy frecuente. Los elementos i,g, y j tienen baja predictibilidad, están asociados a giros. En general, la cadena lineal CRF para el alfabeto PB se le dificulta discriminar entre los elementos asociados a hojas y giros.

Para el alfabeto $SA_{10,3}$, el elemento con la mejor predictibilidad es b asociado a hélices, y el más frecuente. Los elementos d y f tienen una predictibilidad regular, están asociados a hélices y giros. El segundo mejor elemento es e asociado a hojas, no es muy frecuente y su predictibilidad no es tan buena. El elemento i asociado a hojas y el segundo más frecuente, tiene baja predictibilidad. Los elementos con la menor predictibilidad son a, c, h, y j asociados a giros y hojas. En general, la cadena lineal CRF para el alfabeto $SA_{10,3}$ se le dificulta discriminar entre los elementos asociados a hélice-giros y hojas-giros.

Para ambos alfabetos es difícil discriminar los elementos estructurales asociados a giros y hojas, y es más fácil predecir los elementos asociados a hélices.

3.3.4. Importancia de la modelación de la cadena lineal CRF

Describir la estructura 3D de las proteínas por medio de un alfabeto estructural, permite que la cadena lineal CRF modele de forma sencilla e intuitiva las proteínas. En el modelo

	Precisión		Sensibilidad		MCC	
Etiqueta	**Media**	**SD**	**Media**	**SD**	**Media**	**SD**
a	0.5279	0.0058	0.6481	0.0034	0.5671	0.0058
b	0.4407	0.0110	0.2044	0.0041	0.2791	0.0070
c	0.5188	0.0055	0.3689	0.0050	0.3959	0.0056
d	0.6459	0.0055	0.7508	0.0035	0.6271	0.0055
e	0.4565	0.0101	0.4222	0.0133	0.4269	0.0110
f	0.5394	0.0057	0.4471	0.0044	0.4595	0.0052
g	0.3008	0.0193	0.0943	0.0045	0.1633	0.0095
h	0.4273	0.0069	0.4207	0.0139	0.4113	0.0098
i	0.3599	0.0099	0.4545	0.0142	0.3941	0.0108
j	0.4024	0.0348	0.1020	0.0046	0.1991	0.0108
k	0.5836	0.0066	0.5281	0.0032	0.5314	0.0049
l	0.5842	0.0081	0.4790	0.0036	0.5062	0.0067
m	0.7963	0.0038	0.8996	0.0023	0.7772	0.0017
n	0.5347	0.0107	0.6441	0.0078	0.5786	0.0091
o	0.5783	0.0047	0.6081	0.0063	0.5824	0.0058
p	0.4933	0.0089	0.5148	0.0057	0.4864	0.0089

Tabla 3.13: Precisión, sensibilidad, y MCC para cada elemento estructural del alfabeto PB.

CRF, una proteína se entiende como una secuencia de elementos estructurales conectados por los elementos adyacentes, estos elementos son las etiquetas ocultas. Los aminoácidos corresponden a las observaciones y se relacionan con su elemento estructural correspondiente. La ventaja de la cadena lineal CRF con otros modelos secuenciales como un HMM, es que se puede incluir características externas de las observaciones. En los experimentos llevados a cabo, las características externas de los aminoácidos son las etiquetas y puntajes de la estructura secundaria predicha, información PSSM, y características físico-químicas de los aminoácidos.

3.3.5. Comparación con trabajos relacionados

En [42] se menciona que es difícil comparar con precisión diferentes estudios de predicción de estructura local debido a que usan diferentes definiciones de estructura local, diferentes conjuntos de datos, y diferentes criterios de predicción. En esta comparación se pretende resaltar las ventajas y desventajas de las cadenas lineales CRFs frente a otros trabajos de predicción.

Estudios previos en predicción de elementos estructurales, mejoraron sus resultados de predicción basándose en la idea de explotar la relación estructura-estructura, proveniente de avances en la estructura secundaria [23, 56, 75, 12]. La estrategia se basa en usar dos capas: la primera capa utiliza la información de los aminoácidos para predecir la estructura local (relación secuencia-estructura), la segunda capa toma las predicciones de la primera para dar la predicción final de estructura local (relación estructura-estructura). En los trabajos citados, las capas son independientes y utilizan ventanas de tamaño fijo para

	Precisión		Sensibilidad		MCC	
Etiqueta	**Media**	**SD**	**Media**	**SD**	**Media**	**SD**
a	0.2964	0.0091	0.1060	0.0015	0.1525	0.0037
b	0.7332	0.0054	0.9324	0.0005	0.7505	0.0035
c	0.3481	0.0055	0.1734	0.0046	0.2262	0.0057
d	0.5325	0.0042	0.5132	0.0023	0.4709	0.0043
e	0.5728	0.0064	0.5690	0.0028	0.5521	0.0069
f	0.5263	0.0057	0.5572	0.0002	0.5252	0.0077
g	0.4821	0.0047	0.4868	0.0006	0.4293	0.0048
h	0.4558	0.0076	0.3805	0.0049	0.3737	0.0062
i	0.5426	0.0020	0.5068	0.0032	0.4537	0.0031
j	0.4445	0.0021	0.4013	0.0036	0.3644	0.0038

Tabla 3.14: Precisión, sensibilidad, y MCC para cada elemento estructural del alfabeto $SA_{10,3}$.

tomar información de entrada al predictor. En dong et al. [23] utilizan una estrategia de dos capas para predecir el alfabeto PB con una red neuronal y una máquina de soporte vectorial. Tomando como entrada al predictor la matriz PSMM, ambos modelos obtienen resultados Q_{16} similares, 58.5 % para la red neuronal y 58.2 % para la máquina de soporte vectorial. La cadena lineal CRF en el $Exp4$ con la información de los aminoácidos y la matriz PSSM obtiene 58.12 %. Los resultados son cercanos, aunque el CRF además de la matriz PSMM está utilizando la información de los aminoácidos. La ventaja de la cadena lineal CRF es que explota las relaciones estructura-estructura, modelando las relaciones entre elementos estructurales adyacentes, y las relaciones secuencia-estructura con las observaciones y características externas en un sólo *framework* probabilístico.

La información PSSM se caracteriza por obtener buenos resultados en modelos donde se hacen transformaciones no lineales [56, 75, 22, 7]. En [7] se compara una máquina de soporte vectorial SVM y un modelo de regresión logística que reciben como entrada la matriz PSSM para predecir estructura local con el alfabeto estructural PB, cabe anotar que el modelo de regresión logística hace una combinación lineal de la información de entrada y la SVM hace una transformación no lineal con una función kernel de base radial. Los resultados del SVM son superiores a los de la regresión logística. Teniendo en cuenta que la cadena lineal CRF es la versión secuencial de una regresión logística [60]. Conlleva a poner en contexto la debilidad de la cadena lineal CRF para tratar con información evolutiva de la matriz PSSM.

3.4. Conclusiones

Predecir estructura local de las proteínas a partir de la información de los aminoácidos es un problema complejo. Aquí se estudia la capacidad del modelo gráfico lineal CRF con diferentes funciones características en la predicción de estructura local de proteínas con dos alfabetos estructurales PB y $SA_{10,3}$. La cadena lineal CRF empleado en la predicción de elementos estructurales permite explotar las correlaciones de los elementos estructurales

vecinos e integrar la información local de las observaciones de aminoácidos.

Se realizó experimentos con la siguiente información de los aminoácidos: las observaciones de los aminoácidos, matriz PSSM, etiquetas predichas de tres estados de estructura secundaria, puntajes predichos de tres estados de estructura secundaria, características físico-químicas de los aminoácidos, y asignación de la estructura secundaria con DSSP. Cuando se utiliza la información de los aminoácidos se obtiene una exactitud Q_k baja para ambos alfabetos de 42.28 % para PB y 37.31 % para $SA_{10,3}$. Agregar la información de la predicción de estructura secundaria mejora la exactitud en 19.82 % para PB y 18.76 % para $SA_{10,3}$, se obtienen mejores resultados con los puntajes predichos que las etiquetas predichas. Agregar la información de la matriz PSSM a las observaciones de los aminoácidos mejora la exactitud en 15.84 % para PB y 14.09 % para $SA_{10,3}$. Utilizar la información de los aminoácidos, la matriz PSSM, y el puntaje predicho de la estructura secundaria en conjunto proporciona la mejor exactitud con 64.63 % para PB y 59.03 % para $SA_{10,3}$.

Se utilizó la asignación de estructura secundaria para averiguar el límite máximo de exactitud en la predicción. Al utilizarse con la observación de los aminoácidos y la matriz PSSM alcanza un límite máximo Q_k de 70.58 % para PB y 64.25 % para $SA_{10,3}$. La cadena lineal CRF con la predicción de estructura secundaria, la observación de los aminoácidos, y la matriz PSSM obtiene una exactitud cercana al límite máximo, lo que indica que las cadenas lineales CRF aprenden a corregir el error de la predicción de estructura secundaria. Por otro parte, las características físico-químicas no proporcionan buenos resultados al ser usadas en la predicción de los alfabetos estructurales evaluados.

Se realizó un análisis de relación entre los alfabetos PB y $SA_{10,3}$ con los estados de estructura secundaria definidos por DSSP. Se observa que existen elementos asociados exclusivamente a los diferentes tipos de estructura secundaria. Pero hay elementos que son frecuentes en varios estados de estructura secundaria. Para el alfabeto PB se distingue una relación frecuente entre elementos asociados a hojas y giros. Y para el alfabeto $SA_{10,3}$ una relación entre elementos asociados a hélices-giros y hojas-giros. En cuestión de predicción, la mayoría de los elementos más difíciles de predecir para ambos alfabetos son frecuentes en hojas y giros, y es más fácil predecir los elementos asociados a hélices.

El alfabeto estructural que obtiene los mejores resultados de predicción es PB, este alfabeto es muy utilizado para evaluar predictores de alfabetos estructurales por su característica de alta predictibilidad. Con el alfabeto $SA_{10,3}$ se obtienen resultados de predicción más bajos que PB, se cree que esto se debe a las altas frecuencias entre elementos asociados a hélices-giros y hojas-hilos, lo que dificulta discriminarlos.

La cadena lineal CRF como su nombre lo indica hace una combinación lineal de las funciones características y no hace transformaciones no lineales. El uso de la información PSSM se caracteriza por obtener buenos resultados en modelos donde se hacen transformaciones no lineales. Teniendo en cuenta que la cadena lineal CRF es la versión secuencial de una regresión logística. Conlleva a poner en contexto la debilidad de la cadena lineal CRF para tratar con información evolutiva de la matriz PSSM. Se recomienda en trabajos futuros hacer uso de extensiones de un CRF que haga transformaciones no lineales a la información de entrada. Por otra parte, aunque el modelo CRF lineal explota las correlaciones vecinas de los elementos estructurales, sigue siendo un reto abstraer patrones de interacciones de largo rango de los aminoácidos.

Predecir estructura local con elementos estructurales de un alfabeto, es un problema muy similar a la predicción de estructura secundaria. Los métodos usados en la predicción de estructura secundaria también pueden ser usados en la predicción de elementos estructurales. Aunque la predicción de elementos estructurales es útil cuando se necesita la descripción de la estructura 3D de una proteína, como en la predicción de pliegues.

Capítulo 4

Predicción de alfabetos estructurales usando un campo neuronal condicional

Predecir elementos estructurales a partir de la secuencia de aminoácidos es un problema específico del etiquetamiento de datos secuenciales. Etiquetar datos secuenciales radica en tomar como entrada una secuencia de observaciones e inferir un estado para la secuencia (Secuencia de salida), donde el estado es una etiqueta o una segmentación. La secuencia de salida puede ser una estructura compleja. Este tipo de problemas también son conocidos en la comunidad de aprendizaje de maquina como predicción estructurada. Un ejemplo es la predicción de elementos estructurales donde a partir de la secuencia de aminoácidos o sus propiedades se predicen elementos estructurales que representan estructuras complejas en 3D. Lo anterior aplica también para predecir la estructura secundaria de una proteína. El etiquetamiento de datos secuenciales tiene aplicaciones en una gran variedad de áreas como por ejemplo: procesamiento del lenguaje natural, visión por computador y bioinformática. En general, los primeros trabajos de etiquetamiento de secuencias con alfabetos estructurales usaron el modelo oculto de Markov HMM [9], que es un modelo genérico que calcula la probabilidad conjunta de observaciones y etiquetas. Posteriormente y sin orden de ocurrencia se aplicaron modelos discriminatorios como arboles de decisión y bosques aleatorios [57], regresión logística [3], redes neuronales [23, 12], campos aleatorios condicionales CRFs (Capítulo 3) y máquinas de soporte vectorial SVM [56, 75, 7, 57].

Por otra parte, los aspectos centrales en la predicción de elementos estructurales encontrados en el estado del arte son la capacidad de relacionar los aminoácidos con la estructura local y relacionar la estructura-estructura. El último se refiere a encontrar las relaciones que hacen que una proteína tome su estructura nativa global, siendo el más complicado de resolver en los modelos hasta la fecha actual. Este aspecto también hace parte de la predicción de mapas de contacto (predicción de contactos no locales o contactos de largo rango), ya que experimentalmente se evidencia menor precisión para predecir contactos a largo rango, y permanece como un problema abierto.

La predicción de estructura de las proteínas ha mejorado a través de la historia. Un primer

hito fue el uso de información evolutiva PSSM como entrada en modelos de aprendizaje de máquina, propuesto en [29] para predecir la estructura secundaria. Un segundo hito es el uso de varios clasificadores para ir refinando la predicción, como por ejemplo el uso de dos etapas de clasificadores, donde el primero halla relación secuencia-estructura, el segundo toma como entrada los resultados del primero y halla relaciones estructura-estructura [23, 56, 75, 12].

En el capítulo 3, se usó una cadena lineal CRF para predecir elementos estructurales, que explota la información de la secuencia para relacionarla con la estructura local, y la correlación existente de elementos estructurales con la relación de primer orden entre sus etiquetas adyacentes. La ventaja de los CRFs comparada con otros clasificadores como redes neuronales o máquinas de soporte vectorial es que explota directamente la interdependencia entre estructuras locales de residuos adyacentes. Los CRFs son modelos log-lineales ya que sus funciones potenciales son combinaciones lineales de características, pueden modelar bien la correlación entre estructuras locales adyacentes, pero no pueden modelar fácilmente relaciones no lineales entre las características observadas de las proteínas (perfiles PSSM y estructura secundaria), como se observa en los resultados del capítulo 3. Al contrario, las redes neuronales pueden modelar las relaciones no lineales entre las características observadas de las proteínas pero no pueden modelar fácilmente la correlación entre las estructuras locales. Existe un modelo llamado Campo Neuronal Condicional CNF (Conditional Neural Fields) que combina las ventajas de los CRFs y las redes neuronales. Este modelo permite modelar las relaciones implícitas no lineales de características de entrada, y así capturar relaciones más complejas entre la entrada y la salida. También, modela la correlación entre estructuras adyacentes, y tiene las fortalezas de los CRFs como un riguroso modelo probabilístico [53].

A continuación, se explora el poder de los CNF para predecir elementos estructurales con dos alfabetos estructurales (PB y $SA_{10,3}$) dada la información de la secuencia para modelar relaciones complejas secuencia-estructura local y estructura-estructura. Y se evalúa la capacidad de diferentes características de los aminoácidos para mejorar la predicción de estructura local.

4.1. Cadena campo neuronal condicional CNF

En el problema de etiquetar secuencias, se tienen la secuencias de entrada X y salida Y. En la cadena lineal CRF la probabilidad condicional de la secuencia de salida Y dada la secuencia de entrada X es la normalización del producto de exponentes de factores en la cadena.

$$p(Y \mid X) = \frac{1}{Z(X)} exp\left(\sum_{t=1}^{T} \psi(Y, X, t) + \phi(Y, X, t)\right) \tag{4.1}$$

Como se especifica en el capítulo 2, los factores ψ y ϕ corresponden a los factores que relacionan las etiquetas adyacentes y la etiqueta con la observación local, respectivamente. Donde los factores se factorizan con funciones características.

$$\phi(Y, X, t) = \sum_{y} \omega_y^T f_y(Y, X, t) \tag{4.2}$$

$$\psi(Y, X, t) = \sum_{y,y'} \lambda_{y,y'} f_{y,y'}(Y, X, t) \tag{4.3}$$

Aunque, los CRFs han sido ampliamente utilizados y tiene buenos resultados en problemas donde existen relaciones lineales entre características de entrada y etiquetas de salida, no trabajan muy bien en problemas que involucran relaciones más complejas. Algunas áreas donde se presenta lo anterior son computación visual y bioinformática, muchos de estos problemas requieren modelar relaciones no lineales entre entrada y salida [53]. Por ejemplo, si se utilizara una cadena lineal CRF para predecir estructura secundaria de tres estados, para modelar la correlación entre etiquetas y observaciones, las funciones características enumerarían todas las posibles combinaciones entre los estados de la estructura secundaria y la identidad de los aminoácidos. Si se utiliza una gran cantidad de características para crear características más complejas, se pueden generar varias cuestiones: Puede resultar una combinación demasiado grande, y esto ocasiona que la complejidad del modelo crezca. Es más difícil de entrenar el modelo sin que se sobreentrene. Además la enumeración de una gran cantidad de características puede introducir características innecesarias, que incrementan el tiempo de ejecución de la probabilidad.

La cadena Campo Neuronal Condicional CNF (*Conditional Neural Fields*) propuesta por Peng y Bo en [53] no sólo puede parametrizar la probabilidad condicional como los CRFs, sino que también es capaz de modelar implícitamente relaciones no lineales entre la salida y la entrada. En una cadena lineal CNF el factor que relaciona las etiquetas adyacentes es similar al del CRF. Este factor describe la relación entre los vértices de salida adyacentes. Sin embargo el factor que relaciona la etiqueta con la observación local es diferente al CRF. Esta función se define así:

$$\phi(Y, X, t) = \sum_{y} \sum_{g=1}^{K} \omega_{y,g} h\left(\theta_g^T f\left(X, t\right)\right) \delta\left(y_t = y\right) \tag{4.4}$$

$$p(Y \mid X) = \frac{1}{Z(X)} exp\left(\sum_{t=1}^{T} \sum_{y,y'} \lambda_{y,y'} f_{y,y'}(Y, X, t) + \sum_{t=1}^{T} \sum_{y} \sum_{g=1}^{K} \omega_{y,g} h\left(\theta_g^T f\left(X, t\right)\right) \delta\left(y_t = y\right)\right) \tag{4.5}$$

La función h es la función logística. La mayor diferencia entre CRF y CNF radica en el factor ϕ. En CRF es una combinación lineal de funciones características. En CNF hay una capa extra entre la entrada y la salida, que consiste de K funciones compuerta. Las compuertas son extractores de características ocultas no lineales entre la entrada y la salida del modelo en cada posición. En la figura 4.1 se muestra el gráfico correspondiente a un CNF factorizado en la ecuación 4.5.

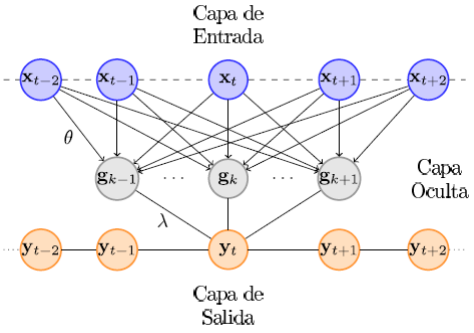


Figura 4.1: Estructura gráfica del modelo Campo Neuronal Condicional.

Comparación entre una cadena lineal CRF y un CNF

A continuación se hablará de los resultados empíricos encontrados en un trabajo sobre el poder que tienen las transformaciones no lineales en los modelos secuenciales. En [64] el modelo CRF es comparado con el modelo de arquitectura profunda (*deep architecture*) para procesamiento de lenguaje natural de Collobert et al. [15] llamado redes neuronales probabilísticas a nivel de oración SLNN (*Sentence-level Likelihood Neural Nets*). La mayoría de modelos desarrollados para etiquetamiento de datos secuenciales emplean generalizaciones de modelos estadísticos lineales. En donde los datos se describen como combinaciones de funciones bases lineales, como las SVM que describen el espacio de las variables de entrada en combinaciones lineales o los CRFs que hacen transformaciones de la distribución de probabilidad (modelos log-lineales). El modelo SLNN presentó nuevos cambios: extendió el modelo de una arquitectura lineal a una no lineal; y reemplazó representaciones de características discretas con representaciones de características distribucionales en un espacio continuo. La comparación entre el modelo CRF y el SLNN se realizó con el fin de responder la pregunta ¿se puede obtener ganancia introduciendo no linealidad a los modelos convencionales basados en características?.

En [64] se resalta que el modelo SLNN fue previamente introducido en la literatura. Se llama Cadena Lineal Campo Neuronal Condicional CNF por Peng y Bo en [53], y posteriormente campos aleatorios condicionales neuronales (*Neural Conditional Random Fields*) por Do y Artieres en [21]. Además, el modelo fue introducido en la literatura de reconocimiento de voz en [54].

La comparación entre el modelo CRF y SLNN se realiza con la evaluación empírica en dos tareas de procesamiento de lenguaje natural: reconocimiento de entidades y fragmentación sintáctica. Para la tarea de reconocimiento de entidades, entrenan los modelos con el conjunto de datos CoNLL-2003, que es un conjunto de artículos de la agencia de noticias Reuters, anotada con cuatro diferentes tipos de entidades: Persona, Lugar, Organización y Misceláneo. Se adapta la notación BIO2, posiciones iniciales e intermedias de una entidad se etiquetan con B- e I-, y las que no sean entidades con la etiqueta O. Los modelos se evalúan en tres diferentes conjuntos de pruebas. Para la tarea de fragmentación sintética, entrenan los modelos con el conjunto CoNLL-2000 y se evalúan en un sólo conjunto de prueba. Los modelos son entrenados con el algoritmo L-BFGS, el modelo SLNN usa 300

unidades ocultas en sus capas ocultas, y se evalúan con las métricas estándares de procesamiento de lenguaje natural: precisión, sensibilidad y puntaje F1. Para los experimentos con representaciones de características en un espacio continuo, toman las representaciones de palabras embebidas (*word embeddings*, que es la representación de una palabra o posiblemente una frase con un vector de números reales en un espacio con bajas dimensiones) usadas en Collobert *et al.* [15], que fueron entrenadas durante alrededor de dos meses con los textos de Wikipedia.

Realizan un experimento para evaluar los modelos cuando se utilizan las representaciones discretas, abordando la pregunta ¿en un espacio de características discretas con altas dimensiones, el modelo SLNN sobrepasa a el modelo CRF?. Para ambas tareas obtienen que el modelo SLNN disminuye el rendimiento en 1 %, asumen el supuesto que se debe al problema de optimización no convexo que tiene que resolver para hallar sus parámetros, pues el modelo tiene en el pequeño conjunto de datos 100 millones de parámetros (437905 características por 300 unidades ocultas), debido a la alta dimensionalidad del espacio de las características de entrada. Otro resultado importante es que el modelo SLNN presenta una pequeña ventaja en rendimiento cuando se utilizan espacios dimensionales discretos pequeños, pero son sobrepasados por la cadena lineal CRF en espacios discretos altamente dimensionales (cuando se utilizan más características). También consideran el factor de posibilidad de que no haya mucha relación no lineal para ser capturada en las tareas ejecutadas, ya que en la comunidad de procesamiento de lenguaje las dos tareas se conocen por tener relaciones más simples comparadas con otras tareas como el reconocimiento de objetos en visión por computador.

Evalúan los modelos utilizando las representaciones distribucionales. Remplazan las representaciones discretas por las representaciones continuas de las palabras embebidas. El modelo SLNN trabaja mucho mejor que la cadena lineal CRF con estas representaciones. Lo que sugiere que hay dependencias estadísticas en las representaciones que no pueden ser efectivamente capturadas con transformaciones lineales, pero que si son capturadas por las capas ocultas de la red neuronal. Este resultado coincide con los resultados de predicción de reconocimiento de escritura en [53] y [21]. Cuando se utilizan las palabras embebidas y características discretas en los modelos, ambos obtienen un incremento en rendimiento parecido, pero el CRF sobrepasa por muy poco al SLNN.

Arquitecturas de la combinación de CRFs con Redes Neuronales

Las arquitecturas de los modelos planteados en los trabajos de [53],[54] y [15] son similares. Su arquitectura se puede resumir con la idea de relacionar las etiquetas adyacentes y extraer características no lineales de la entrada con unidades ocultas de una sola capa de una red neuronal. A diferencia de estos trabajos, en do et al. [21] se tienen en cuenta más de una capa de la red neuronal para extraer características ocultas no lineales. La motivación de usar más de una capa oculta, viene de los avances en la capacidad de las arquitecturas de redes neuronales profundas (*deep neural networks*) para extraer características de alto nivel en los datos. Entonces su idea se basa en extraer características relevantes con la red neuronal profunda que sean la entrada para la cadena lineal CRF. Al usar la red para extraer las características, el aprendizaje se vuelve un problema de optimización no convexo. Entrenan el modelo con el algoritmo que revivió el interés en la redes profundas

[27], primero se pre entrena cada capa con un algoritmo no supervisado. Al final del pre entrenamiento se obtienen los parámetros iniciales del modelo. Estos parámetros se utilizan para entrenar la parte del CRF de manera supervisada. Posteriormente todo el modelo es afinado con el algoritmo estándar de propagación hacia atrás.

En [13] se plantea la misma arquitectura del modelo de [21] pero difieren en que se entrenan diferente. Los resultados son buenos, supera el estado del arte de muchas tareas de predicción estructurada en diferentes conjuntos de datos reconocidos. Es así como el uso de una arquitectura profunda para extraer representaciones combinada con el modelo probabilístico CRF proporciona un gran *framework* para predicción de secuencias con salida estructurada.

4.1.1. Algunas aplicaciones del campo neuronal condicional CNF

En [53] evaluaron el modelo CNF para predecir tres estados de estructura secundaria de las proteínas. La observación del modelo es la secuencia de la matriz PSSM, $X = (x_1, x_2, \ldots, x_n)$ donde cada x_i es un vector de 20 elementos, correspondientes a los puntajes del perfil. La salida es $Y = (H, E, C)$ representando la estructura secundaria para el i residuo. Obteniendo los mejores resultados con una configuración de 30 unidades ocultas y una ventana de tamaño 13. Los resultados del CNF se comparan con una cadena lineal CRF y redes neuronales usando el conjunto de datos CB513. También lo comparan con métodos generales de predicción como: Semi-Markov HMM, SVMpsi, PSIPRED, YASSPP y SPINE, en donde algunos toman otras características como entrada. El CNF obtiene la mejor precisión entre todos los métodos, superando ampliamente a la cadena lineal CRF, confirmando la relación no lineal entre la secuencia de perfiles y la salida de la estructura secundaria. El CNF al tener en cuenta las relaciones entre estructuras adyacentes, obtiene mejor predicción que las redes neuronales. Además se evalúa el modelo CNF en el reconocimiento de escritura utilizando el conjunto de datos OCR, que contiene 6876 secuencias. En OCR cada palabra consiste de una secuencia de caracteres y cada carácter es una imagen de $16x8$ pixeles binarios. La entrada $X = (x_1, x_2, \ldots, x_n)$ es una secuencia de vectores de 128 dimensiones. La salida que se quiere predecir es la secuencia de etiquetas donde una etiqueta es una de 26 clases posibles $\{a, b, \ldots, z\}$. Obtienen el mejor resultado con 40 unidades ocultas y un tamaño de ventana de 1. El CNF supera ampliamente a la cadena lineal CRF, redes neuronales y máquinas de soporte vectorial.

Wang et al. en [67] usan un CNF para predecir ocho estados de estructura secundaria (DSSP). Este problema es más difícil de predecir con técnicas de aprendizaje de maquina comparado con la predicción de tres estados, por su distribución no balanceada de los 8 estados en las estructuras nativas. A la fecha del trabajo de Wang et al. no existían muchos predictores de ocho estados, el mejor era SSPro8. Para predecir, toman diferentes características de los aminoácidos como entrada al modelo y realizan diferentes experimentos para evaluar el efecto de su uso en la predicción. Las características utilizadas son: información PSSM, propensión de ser un punto extremo en la estructura secundaria, propiedades físico-químicas, potenciales de contacto relacionados de los aminoácidos, y la secuencia primaria (los aminoácidos). Para evaluar la capacidad de predicción del modelo utilizan tres conjuntos de datos: CullPDB, CB513 y RS126. Primero realizan validación cruzada de cinco iteraciones en el conjunto de datos CB513 y obtienen una precisión Q_8 de 63.3 %,

envían al servidor SSPro8 este conjunto y obtienen un Q_8 de 51 %. Después, se realiza validación cruzada de cinco iteraciones con CullPDB, y prueban en el conjunto CB513 y RS126. El modelo CNF obtiene una precisión Q_8 de 64.9 % y 64.7 %, SSPro8 obtiene 51 % y 48 %, en los conjuntos de prueba respectivos. Por otra parte, realizan experimentos con diferentes características de entrada al modelo para evaluar el efecto en la predicción. La característica más importante para predecir es la información evolutiva PSSM. Sin usar la información de la matriz PSSM, la característica propensión de ser un punto extremo en la estructura secundaria funciona mejor que las demás. Sin embargo cuando se integra la información PSSM no hay una diferencia clara en el efecto de las demás características.

Zhao et al. en [74] utilizan un CNF en el proceso de predicción de plegamientos de proteínas. Específicamente su método consiste en crear muestras de fragmentos estructurales de forma continua, para posteriormente acoplado con una función de energía y simulación Monte Carlo de intercambio de réplicas (REMC) producir estructuras señuelos o *decoys*. Utilizan un representación simplificada de un modelo de una proteína, se asume que la distancia entre dos átomos C_α adyacentes es constante y representa el rastro-C_α usando un conjunto de pseudo ángulos del *backbone* (θ, τ). Cada (θ, τ) corresponde a un vector unitario, descritos por la distribución Fisher–Bingham (FB5). Usando un conjunto de entrenamiento agrupan todos los ángulos (θ, τ) en 100 grupos. Calculan la distribución FB5 para cada grupo con el estimador KentEstimator. Con lo anterior pueden realizar muestreo, y explorar espacios conformacionales continuos de las proteínas. El modelo CNF juega el papel que dada la información PSSM y la estructura secundaria predicha de una secuencia de aminoácidos de una proteína, predice para cada aminoácido en la secuencia el grupo de los pseudo ángulos. El CNF es de orden dos, esto significa que la transición entre estados no sólo depende del estado anterior (vecino) sino que también del anterior al vecino. El modelo logra un puntaje F1 de 23.44 % en una validación cruzada de cinco iteraciones con 200 unidades ocultas y un factor de regularización de 50. Que el valor F1 sea bajo, se atribuye a la gran cantidad de estados utilizados. Aunque la diferencia entre el puntaje F1 de un trabajo previo donde utilizaron una cadena lineal CRF para el mismo problema es poca, la capacidad de muestreo es mejor con el modelo CNF.

En [47] motivados por los errores de alineación que presentan los métodos de modelación de proteínas basados en plantillas, como modelación de homólogos y reconocimiento de pliegues (*protein threading*), especialmente cuando la identidad entre dos secuencias es menor a 30 %. Proponen la herramienta CNFpred para alinear una secuencia con una plantilla relacionada de una forma más precisa. La herramienta enfrenta dos limitaciones de los métodos previos a CNFpred para la alineación de proteínas: primero, el uso de una función de puntaje lineal que guiá la predicción secuencia-alineamiento; segundo, los errores de los métodos cuando los perfiles de las proteínas son muy dispersos. CNFpred combina información de la secuencia e información estructural en una función de puntaje probabilística no lineal. Dada la proteína plantilla T y la proteína objetivo S, el problema consiste en denotar el alineamiento $A = \{a_1, a_2, \ldots a_L\}$ entre T y S, donde L es el tamaño del alineamiento y a_i es uno de tres estados posibles: M (dos residuos en alineamiento), I_t (inserción en la proteína plantilla) y I_s (inserción en la proteína objetivo). A diferencia del modelo CNF convencional [53], el planteado para predecir el alineamiento usa unidades ocultas entre las etiquetas adyacentes. En lugar de entrenar el modelo maximizando la probabilidad de un conjunto de alineamientos referentes construidos por una herramienta de alienación, lo entrena con un método sensitivo a la calidad, que pone más peso en

las áreas más conservadas para asegurar alineación correcta. Comparado con diferentes métodos CNFpred los supera, incluso cuando la información de la secuencia es dispersa.

Ma y Wang en [48] presentan la herramienta AcconPred para predecir accesibilidad al solvente y número de contactos de manera simultánea para una proteína, basado en un *framework* de aprendizaje multitarea de compartir parámetros bajo un CNF. La accesibilidad al solvente de un residuo de una proteína es el área de la superficie que es accesible a un solvente. Esta juega un rol importante en el proceso de plegamiento ya que está relacionada a la configuración espacial y envoltura de la proteína. Por otra parte el número de contacto de un residuo es la información resultante del plegamiento. Aunque las dos propiedades son diferentes, están ciertamente relacionadas, ambas reflejando la atmósfera hidrofílica o hidrofóbica de un residuo en la estructura de la proteína. Para predecir accesibilidad al solvente se utilizan tres estados o etiquetas B (*buried*), I (*intermediate*) e E (*exposed*). El número de contacto de un residuo es el número de átomos C-beta de los otros residuos en una esfera de radio 7.5 Å centrada en el átomo C-beta del i residuo, el número máximo de contactos está limitado a 14, así tiene 15 estados o etiquetas. El *framework* multitarea comparte los pesos de las unidades ocultas que hacen las transformaciones no lineales en el CNF. Toman como entrada al modelo información PSSM, predicción de estructura secundaria, propensión de ser un punto extremo en la estructura secundaria, propiedades físico-químicas y potenciales de contacto correlacionados de los aminoácidos. Entrenan y validan el modelo con un conjunto de datos de 5729 proteínas monomérico, globular, y estructuras de proteínas no de membrana. El modelo obtiene 68 % Q_3 para accesibilidad al solvente y 30 % Q_{15} para número de contactos. El CNF bajo el *framework* multitarea aprende representaciones genéricas para ambos en la capa de transformaciones no lineales y tiene una ganancia en la predicción de 2 %, además explota las relaciones entre estados adyacentes.

4.2. Especificaciones del campo neuronal condicional CNF

El CNF será aplicado al problema de predicción de la estructura local mediante alfabetos estructurales, de la siguiente manera: Al igual que la aplicación del problema con cadenas lineales CRFs, la secuencia de observaciones X corresponderá a los aminoácidos $X = \{x_1, x_2, \ldots, x_n\}$, n es el tamaño de la proteína. La secuencia de etiquetas Y son los elementos estructurales de un alfabeto y corresponde a los elementos que codifican la proteína $Y = \{y_1, y_2, \ldots, y_m\}$, m es el número de elementos estructurales necesarios para codificarla, y $y_i \, \epsilon \{1, 2, \ldots, k\}$ donde k el número total de elementos estructurales en el alfabeto.

En un *framework* probabilístico, el modelo explotará las relaciones adyacentes entre los elementos estructurales. Y mediante la capa de la red neuronal, el modelo extraerá relaciones no lineales entre la información de los aminoácidos y su estructura local.

4.2.1. Entrenamiento y predicción

A continuación se describe la inferencia y entrenamiento establecido por [53] para los CNFs. Los parámetros de la capa oculta y los del CRF pueden ser optimizados de manera

conjunta. Para realizar inferencia en el modelo, después de haberse aprendido los parámetros, primero se calculan los valores de la capa oculta dada la entrada y se procede a usar los algoritmos de inferencia para predecir la salida. Cualquier algoritmo de inferencia para cadenas lineales CRFs puede ser utilizado en CNF. En el entrenamiento del CNF se utiliza *forward-backward*. Para predecir las etiquetas más probables dada una secuencia de observaciones, se puede usar Viterbi o al igual que la cadena lineal CRF, la probabilidad marginal más alta para cada etiqueta.

Para entrenar el modelo, al igual que la cadena lineal CRF se maximiza la log verosimilitud. Dado un conjunto de entrenamiento $D = \left\{x^{(i)}, y^{(i)}\right\}_{i=1}^{N}$, donde cada $x^{(i)} = \left\{x_1^{(i)}, x_2^{(i)}, \ldots, x_T^{(i)}\right\}$ es una secuencia de entrada y cada $y^{(i)} = \left\{y_1^{(i)}, y_2^{(i)}, \ldots, y_T^{(i)}\right\}$ es una secuencia de etiquetas, se maximiza $\ell(\theta)$.

$$log\, p\,(Y|X) = \sum_{t=1}^{T} \left(\psi\,(Y, X, t) + \phi\,(Y, X, t)\right) - log\, Z(x)$$

$$\ell\,(\theta) = \sum_{n=1}^{N} log\, p\,(Y_n|X_n)$$

Dado que el CNF contiene una capa que hace una transformación no lineal (la función h), la log verosimilitud no es convexa. Por lo tanto es probable que se obtenga una solución local de los parámetros. Para entrenar los parámetros de la capa oculta y el CRF conjuntamente se usa optimización basada en gradiente. De la misma forma que la cadena lineal CRF, el CNF utiliza LBFGS como la rutina de optimización para encontrar los parámetros. En el entrenamiento del CNF se usa regularización L2, se tienen tres vectores de parámetros (los correspondientes a el factor de transición, el factor entre la capa oculta y la etiqueta, y los de la capa oculta) que tienen que tenerse en cuenta en la regularización. Para mayor detalle sobre el proceso de entrenamiento remitirse a [53].

4.2.2. Características de entrada al modelo

La información de entrada al modelo corresponde a información de los aminoácidos y puntajes de predicción de la estructura secundaria.

Se utiliza la codificación one-hot por aminoácido en la secuencia, esto es un vector con 20 dimensiones para representar cada aminoácido. Cada aminoácido en la secuencia se representa con su vector de 20 dimensiones donde tiene un 1 para el aminoácido correspondiente y un 0 para los demás. Se usa la matriz PSSM (*position-specific scoring matrix*) que contiene información evolutiva, se deriva de secuencias de homólogos y es dependiente de la posición en la secuencia. Es de tamaño $n \times 20$, donde n es el tamaño de la proteína y proporciona un puntaje de sustitución para cada uno de los aminoácidos en cada posición. Se obtiene por medio del programa de alineación PSI-BLAST con cinco iteraciones y E-value = 0.001, utilizando una base de datos no redundante NR. Previamente la base de datos fue filtrada, removiendo secuencias con baja información y regiones de hélice superenrollada con el programa pfilt.

La información estructural usada como entrada corresponde a los puntajes de la estructura secundaria predicha por PSSPRED [68]. También, se usa la información físico-química de los aminoácidos definida por [50], que define para cada aminoácido los valores de parámetro estérico, polarizabilidad, volumen, hidrofobicidad, y punto isoeléctrico.

4.3. Materiales y experimentos

La metodología y materiales usada para llevar a cabos los experimentos corresponden a la misma definida previamente en el capítulo 3 sección 3.2. Los resultados de los experimentos se obtienen por validación cruzada de cinco iteraciones sobre el conjunto de datos; 4/5 de los datos se utilizan para entrenar y 1/5 de los datos para validar. El conjunto de datos es de 3052 secuencias y estructuras de proteínas no homólogas. Los alfabetos usados son PB y $SA_{10,3}$. Y la evaluación de los resultados se lleva a cabo con la exactitud Q_k, precisión, sensibilidad, y coeficiente de correlación de Matthews MCC.

4.4. Resultados y análisis

4.4.1. Experimentos con diferentes características de las proteínas

Se llevan a cabo experimentos que combinan la información de los aminoácidos para identificar las características de entrada que mejoran la predicción de alfabetos estructurales.

4.4.1.1. Selección del tamaño de ventana y número de neuronas

Al igual que la cadena lineal CRF, los resultados del modelo CNF se ven afectados por el tamaño de la ventana de contexto y el parámetro de regularización en el entrenamiento. Pero además, el CNF se afecta por el número de neuronas de la capa oculta. A continuación se realizan experimentos con el mismo parámetro de regularización, variando el tamaño de la ventana de contexto y el número de neuronas. Para seleccionar los parámetros de tamaño de ventana y número de neuronas con los que se obtenga buen rendimiento Q_k.

Todos los experimentos usan la matriz PSSM como entrada al modelo y el parámetro de regularización $\lambda = 1$. Las tablas 4.1 y 4.2 contienen los resultados Q_k para diferentes tamaños de ventanas para el alfabeto PB y $SA_{10,3}$, respectivamente. Los valores de tamaño de ventana y número de neuronas seleccionados para ambos alfabetos es de tamaño 9 y 60 neuronas con un Q_{16} de 63.72 % para PB y Q_{10} de 59.39 % para $SA_{10,3}$.

	Número de neuronas	
Tamaño ventana	**G40**	**G60**
7	63.29	63.60
9	63.33	63.72
13	63.31	63.50

Tabla 4.1: Rendimiento Q_{16} por tamaño de ventana y número de neuronas para el alfabeto PB.

	Número de neuronas	
Tamaño ventana	**G40**	**G60**
7	59.04	59.16
9	59.31	59.39

Tabla 4.2: Rendimiento Q_{10} por tamaño de ventana y número de neuronas para el alfabeto $SA_{10,3}$.

4.4.1.2. Experimentos con información de los aminoácidos y predicción de la estructura secundaria

Se conforman cuatro experimentos con información de los aminoácidos y predicción de la estructura secundaria. La tabla 4.3 muestra las combinaciones de las características utilizadas en cada experimento. La descripción de las características usadas, se encuentran en la sección 4.2.2. Los resultados Q_k de las predicciones de los CNFs para el alfabeto PB y $SA_{10,3}$ con diferentes valores del parámetro de regularización se muestran en las tablas 4.4 y 4.5, y en la figura 4.2.

El experimento base es $Exp1$, solamente usa codificación one-hot para aminoácidos. El experimento $Exp2$ agrega al experimento base los puntajes predichos de la estructura secundaria. El experimento $Exp3$ agrega al experimento base la matriz PSSM. Y el experimento $Exp4$ utiliza las tres características de entrada al modelo.

Al igual que la cadena lineal CRF se usa la regularización L2 para prevenir el sobre entrenamiento. Debido a que encontrar el mejor factor de regularización requiere computación intensiva, se evalúan cuatro factores de regularización $\lambda = \{1, 4, 9, 19\}$.

Entrada al modelo	$Exp1$	$Exp2$	$Exp3$	$Exp4$
Codificación one-hot para aminoácidos	+	+	+	+
Puntajes de la predicción de las estructuras secundarias	-	+	-	+
Perfil PSSM	-	-	+	+

Tabla 4.3: Configuración de experimentos para evaluar información de los aminoácidos y predicción de la estructura secundaria.

En el siguiente análisis se utilizan los resultados con el parámetro de regularización $\lambda = 9$. El experimento base $Exp1$ obtiene una exactitud de 52.25 % para PB y 48.65 % para $SA_{10,3}$. El experimento $Exp2$ agrega al experimento base los puntajes predichos de estructura secundaria, obtiene una exactitud de 65.45 % para PB y 59.86 % para $SA_{10,3}$. Al comparar su resultado con el experimento base tiene una ganancia de 13.2 % para PB y 11.21 % para $SA_{10,3}$. El experimento $Exp3$ agrega la matriz PSSM al experimento base, obteniendo una exactitud de 66.45 % para PB y 61.44 % para $SA_{10,3}$. Al comparar con

el experimento base se tiene una ganancia de 14.2 % para PB y 12.79 % para $SA_{10,3}$. Se compara los resultados del experimento $Exp3$ y $Exp2$ para evaluar que característica entre la matriz PSSM y los puntajes predichos de estructura secundaria proporcionan el mejor resultado. La información de la matriz PSSM en el experimento $Exp3$ supera al experimento $Exp2$ por 1 % para PB y 1.58 % para $SA_{10,3}$. El experimento $Exp4$ que utiliza las tres características de entrada proporciona los mejores resultados de exactitud para ambos alfabetos con 68.53 % para PB y 63.29 % para $SA_{10,3}$.

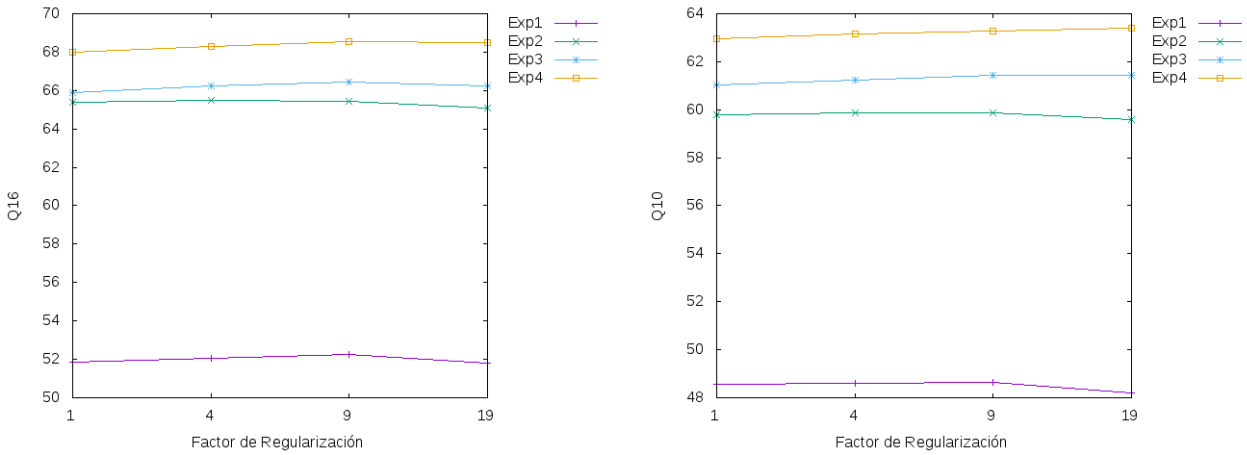
Los factores del parámetro de regularización evaluados muestra que los resultados no se afectan en gran medida por este, pero es necesario para evitar el sobre entrenamiento.

Experimentos	$\lambda = 1$	$\lambda = 4$	$\lambda = 9$	$\lambda = 19$
$Exp1$	51.86	52.08	52.25	51.79
$Exp2$	65.41	65.48	65.45	65.10
$Exp3$	65.88	66.23	66.45	66.24
$Exp4$	67.97	68.29	68.53	68.48

Tabla 4.4: Resultados de la exactitud Q_{16} del alfabeto PB para diferentes valores de regularización.

Experimentos	$\lambda = 1$	$\lambda = 4$	$\lambda = 9$	$\lambda = 19$
$Exp1$	48.56	48.61	48.65	48.21
$Exp2$	59.80	59.86	59.86	59.59
$Exp3$	61.02	61.24	61.44	61.42
$Exp4$	62.97	63.16	63.29	63.38

Tabla 4.5: Resultados de la exactitud Q_{10} del alfabeto $SA_{10,3}$ para diferentes valores de regularización.



(a) (b)

Figura 4.2: Resultados de exactitud de los experimentos con diferentes valores de regularización. (a) Resultado Q_{16} para el alfabeto PB y (b) Resultado Q_{10} para el alfabeto $SA_{10,3}$.

4.4.1.3. Experimentos con información físico-químicas de los aminoácidos

En esta sección se evalúa las características físico-químicas de los aminoácidos en la exactitud de la predicción. Se utiliza las funciones características físico-químicas descrita en la sección 4.2.2. El mejor experimento de la sección anterior es seleccionado para agregarle las características físico-químicas. Se realiza el experimento $Exp4'$, este corresponde al experimento $Exp4$ con las características físico-químicas agregadas. La tabla 4.6 muestra la configuración del experimento y resultados para los alfabetos PB y $SA_{10,3}$ con el parámetro de regularización $\lambda = 9$.

El experimento $Exp4'$ obtiene 68.16 % para PB y 63.06 % para $SA_{10,3}$. Si se compara con los resultados del experimento $Exp4$, se disminuye la exactitud 0.37 % para PB y 0.23 % para $SA_{10,3}$. Por lo tanto se concluye que las características físico-químicas evaluadas no contribuyen a mejorar la exactitud de la predicción.

Entrada al modelo	$Exp4'$
Codificación one-hot para aminoácidos	+
Puntajes de la predicción de las estructuras secundarias	+
Perfil PSSM	+
Propiedades físico-químicas	+
Alfabeto	$Exp4'$
PB	68.16
$SA_{10,3}$	63.06

Tabla 4.6: Configuración y resultados Q_k de experimentos con características físico-químicas para los alfabetos PB y $SA_{10,3}$.

4.4.2. Precisión, sensibilidad, y MCC en el conjunto de datos

A continuación, se realiza un análisis teniendo en cuenta la relación entre elementos estructurales y estructura secundaria de la sección 3.3.1 para resaltar la predictibilidad de los elementos estructurales. Las tablas 4.7 y 4.8 muestran la precisión, sensibilidad, y MCC de la validación cruzada del experimento $Exp4$ con factor de regularización $\lambda = 9$ para los alfabetos estructurales PB y $SA_{10,3}$, respectivamente.

Para PB, de igual forma que la cadena lineal CRF los elementos m, n, o, k, l, y p asociados a hélices son más predecibles. El elemento m tiene la mejor predictibilidad y es muy frecuente, los elementos n, o, k, l, y p tienen buena predictibilidad a pesar de su poca frecuencia. El elemento b asociado a hojas y giros, tienen baja predictibilidad. A diferencia de este, los elementos f y a que también son frecuentes en hojas y giros tienen buena predictibilidad. El elemento d tiene buena predictibilidad, es exclusivo de hojas y muy frecuente. Los elementos g y j son los menos predecibles, están asociados a giros. El elemento i que también está asociado a giros tiene predictibilidad regular.

Para $SA_{10,3}$, el elemento con la mejor predictibilidad es b asociado a hélices, y el elemento más frecuente. De los elementos asociados con hélices-giros, los elementos d y f tienen buena predictibilidad y los elementos a y c tienen baja predictibilidad, el elemento f es el segundo con mejor predictibilidad y es poco frecuente. De los elementos asociados con hojas-giros, los que tienen buena predictibilidad son g e i, y los elementos con baja predictibilidad son j y h. El elemento e asociado a giros tiene buena predictibilidad a pesar de ser poco frecuente.

	Precisión		**Sensibilidad**		**MCC**	
Etiqueta	**Media**	**SD**	**Media**	**SD**	**Media**	**SD**
a	0.6064	0.0090	0.6435	0.0078	0.6095	0.0078
b	0.4852	0.0093	0.3177	0.0064	0.3709	0.0075
c	0.5438	0.0055	0.5050	0.0042	0.4836	0.0047
d	0.6788	0.0068	0.7869	0.0070	0.6670	0.0062
e	0.5028	0.0168	0.4977	0.0078	0.4891	0.0101
f	0.5767	0.0067	0.5316	0.0065	0.5244	0.0071
g	0.3834	0.0124	0.1431	0.0070	0.2291	0.0085
h	0.4802	0.0162	0.4596	0.0131	0.4582	0.0123
i	0.4920	0.0199	0.4354	0.0167	0.4550	0.0164
j	0.3380	0.0147	0.2798	0.0136	0.3019	0.0113
k	0.6044	0.0061	0.5997	0.0066	0.5798	0.0062
l	0.6215	0.0056	0.5324	0.0074	0.5543	0.0064
m	0.8472	0.0038	0.8993	0.0039	0.8145	0.0045
n	0.6805	0.0090	0.6211	0.0152	0.6440	0.0115
o	0.6472	0.0080	0.6214	0.0113	0.6250	0.0090
p	0.5807	0.0047	0.5318	0.0075	0.5410	0.0059

Tabla 4.7: Precisión, sensibilidad, y MCC para cada elemento estructural del alfabeto PB.

	Precisión		Sensibilidad		MCC	
Etiqueta	**Media**	**SD**	**Media**	**SD**	**Media**	**SD**
a	0.4141	0.0115	0.2148	0.0130	0.2733	0.0117
b	0.8229	0.0033	0.9103	0.0025	0.8025	0.0024
c	0.4903	0.0140	0.2303	0.0059	0.3199	0.0084
d	0.5645	0.0048	0.6115	0.0031	0.5396	0.0043
e	0.5898	0.0097	0.6184	0.0065	0.5862	0.0078
f	0.6175	0.0093	0.5723	0.0109	0.5811	0.0100
g	0.5175	0.0073	0.5853	0.0026	0.4988	0.0057
h	0.4906	0.0055	0.4469	0.0064	0.4285	0.0053
i	0.5448	0.0024	0.6005	0.0018	0.5023	0.0017
j	0.4972	0.0107	0.4042	0.0182	0.3972	0.0155

Tabla 4.8: Precisión, sensibilidad, y MCC para cada elemento estructural del alfabeto $SA_{10,3}$.

4.4.3. Comparación entre los resultados del modelo Campo Neuronal Condicional y Campo Aleatorio Condicional CRF

Se compara los resultados con el factor de regularización $\lambda = 9$ de las cadenas lineales CRF y el CNF de acuerdo a la información de entrada al modelo, se tienen entonces las siguientes comparaciones:

Información de los aminoácidos:

La tabla 4.9 contiene los resultados de los experimentos CRF-$Exp1$ y CNF-$Exp1$. Estos experimentos corresponden a las configuraciones con la información de los aminoácidos. Se diferencian en que la cadena lineal CRF usa las observaciones de los aminoácidos y no la codificación one-hot. Al comparar los resultados de la cadena lineal CRF y el CNF, el CNF supera a la cadena lineal CRF por 9.97 % para PB y 11.34 % para $SA_{10,3}$.

El modelo CNF recibe la representación one-hot de los aminoácidos, la capa oculta del CNF utilizando esta representación de bajas dimensiones logra crear características que son más informativas para clasificar los elementos estructurales. Por el contrario, la cadena lineal CRF utiliza muchas funciones características discretas dadas por las observaciones de los aminoácidos en el CRF y no logra superar al CNF.

Modelo/Alfabeto	**Predicción PB** Q_{16}	**Predicción** $SA_{10,3}$ Q_{10}
CRF-$Exp1$	42.28	37.31
CNF-$Exp1$	52.25	48.65

Tabla 4.9: Resultados de la exactitud Q_k para los alfabetos PB y $SA_{10,3}$ del experimento $Exp1$ del modelo CRF y CNF.

Información de los aminoácidos y puntajes de estructura secundaria predicha:

La tabla 4.10 contiene los resultados de los experimentos CRF-$Exp3$ y CNF-$Exp2$. Estos experimentos corresponden a las configuraciones con la información de los aminoácidos y puntajes de estructura secundaria predicha. Al comparar los resultados de la cadena lineal CRF y el CNF, el CNF supera a la cadena lineal CRF por 3.44 % para PB y 3.79 % para $SA_{10,3}$.

La información de la estructura secundaria predicha ayuda a mejorar considerablemente la exactitud en ambos modelos. A pesar que la cadena lineal CRF tiene una exactitud más baja que la del modelo CNF, sus resultados son cercanos. Esto se debe a que la estructura predicha ayuda a complementar la información de los aminoácidos en ambos modelos, pero mucho más en la cadena lineal CRF.

Modelo/Alfabeto	**Predicción PB** Q_{16}	**Predicción** $SA_{10,3}$ Q_{10}
CRF-$Exp3$	62.01	56.07
CNF-$Exp2$	65.45	59.86

Tabla 4.10: Resultados de la exactitud Q_k para los alfabetos PB y $SA_{10,3}$ del experimento $Exp3$ del modelo CRF y experimento $Exp2$ del modelo CNF.

Información de los aminoácidos y matriz PSSM:

En el capítulo 3 en la sección 3.3.5, teniendo en cuenta que una cadena lineal CRF es la versión secuencial de la regresión logística [60]. Se resaltó la debilidad de las cadenas lineales CRFs al tratar con la matriz PSSM basados en la comparación de la SVM y la regresión logística de [7].

La tabla 4.11 contiene los resultados de los experimentos CRF-$Exp4$ y CNF-$Exp3$. Estos experimentos corresponden a las configuraciones con la información de la matriz PSMM y los aminoácidos. Al comparar los resultados de la cadena lineal CRF y el CNF, el CNF supera a la cadena lineal CRF por 8.33 % para PB y 10.04 % para $SA_{10,3}$.

El hecho que la cadena lineal CRF al usar la información PSSM no obtenga buenos resultados, se debe a la naturaleza del modelo. Ya que el uso de la información PSSM se caracteriza por obtener buenos resultados en modelos donde se hacen transformaciones no lineales [56, 75, 22, 7, 67]. Y la cadena lineal CRF hace combinaciones lineales a la información de entrada. La afirmación anterior se soporta con los resultados empíricos del trabajo de Wang y Manning en [64], donde un modelo no lineal es mas efectivo que una cadena lineal CRF con representaciones distribucionales de bajas dimensiones, en este trabajo este tipo de información de entrada al modelo es la matriz PSSM.

Modelo/Alfabeto	**Predicción PB** Q_{16}	**Predicción** $SA_{10,3}$ Q_{10}
CRF-$Exp4$	58.12	51.40
CNF-$Exp3$	66.45	61.44

Tabla 4.11: Resultados de la exactitud Q_k para los alfabetos PB y $SA_{10,3}$ del experimento $Exp4$ del modelo CRF y experimento $Exp3$ del modelo CNF.

Información de los aminoácidos, puntajes de estructura secundaria predicha, y matriz PSSM:

La cadena lineal CRF y el CNF obtienen los mejores resultados con la información de los aminoácidos, los puntajes de la predicción de estructura secundaria, y la matriz PSSM. La configuración de las características anteriores corresponde al experimento $Exp6$ de la cadena lineal CRF y al experimento $Exp4$ del CNF. Estos se diferencian en que la cadena lineal CRF usa las observaciones de los aminoácidos y no la codificación one-hot, pero se consideran equivalentes para comparar sus resultados. La tabla 4.12 contiene los resultados de los experimentos para ambos modelos. Al comparar los resultados de la cadena lineal CRF y el CNF, el CNF supera a la cadena lineal CRF por 3.9 % para PB y 4.03 % para $SA_{10,3}$. La ganancia en la predicción del modelo CNF se debe a la capa oculta que realiza transformaciones no lineales a la información de entrada, esta logra capturar patrones más complejos entre la entrada y la salida del modelo.

Modelo/Alfabeto	**Predicción PB** Q_{16}	**Predicción** $SA_{10,3}$ Q_{10}
CRF-$Exp6$	64.63	59.03
CNF-$Exp4$	68.53	63.29

Tabla 4.12: Resultados de la exactitud Q_k para los alfabetos PB y $SA_{10,3}$ del experimento $Exp6$ del modelo CRF y experimento $Exp4$ del modelo CNF.

Comparación la tabla de precisión, sensibilidad y MCC

El modelo CNF supera a la cadena lineal CRF en la métrica MCC para todos los elementos de ambos alfabetos estructurales. Para el alfabeto PB, al comparar las tablas 4.7 y 3.13. Los elementos que mejoran significativamente son c, e, y f asociados a hojas-giros y los elementos b, g, h, i, y j asociados a giros. Para el alfabeto $SA_{10,3}$, al comparar las tablas 4.8 y 3.14. Los elementos que mejoran significativamente son a, b, c, d, y f asociados a hélice-giros y los elementos g y h asociados a hojas-giros. Lo anterior indica que el modelo CNF con su capa oculta crea características para el alfabeto PB que ayuda a discriminar mejor los elementos asociados a hojas y giros. Y para el alfabeto $SA_{10,3}$ crea características que ayudan a discriminar mejor los elementos relacionados con hélice-giros y hoja-giros.

4.4.4. Comparación con trabajos relacionados

En la sección 3.3.5 del capítulo 3, se menciona que es difícil comparar con precisión diferentes estudios de predicción de estructura local debido a que usan diferentes definiciones de estructura local, diferentes conjuntos de datos, y diferentes criterios de predicción [42]. En esta comparación se pretende resaltar las capacidades de las cadenas CRFs frente a otros trabajos de predicción.

En dong et al. [23] utilizan una estrategia de dos capas para predecir el alfabeto PB, la primera capa toma como entrada al predictor la matriz PSSM, y la segunda capa toma el resultado de la primera para retornar la predicción. Entrenan su modelo con una red neuronal y una máquina de soporte vectorial. Ambos modelos obtienen resultados Q_{16} similares, 58.5 % para la red neuronal y 58.2 % para la máquina de soporte vectorial. Pero la máquina de soporte vectorial tarda mucho mas en entrenar cuando el conjunto de entrenamiento es grande. En [75] introducen la herramienta LOCUSTRA para la predicción del alfabeto PB, utilizan la estrategia de dos capas y como clasificador máquinas de soporte vectorial. La primera capa consiste de 120 clasificadores, que corresponden clasificadores binarios acoplados por pares. Esta capa recibe como entrada la matriz PSSM. La segunda capa consiste de 16 clasificadores, que corresponden a un clasificador por elemento estructural. La segunda capa toma como entrada los resultados de la primera capa. LOCUSTRA reporta una exactitud Q_{16} de 61.0 %.

La cadena lineal CRF con la observación de los aminoácidos y la matriz PSSM en el experimento $Exp4$ obtiene 58.12 %. Este resultado es cercano al de los dos trabajos mencionados, a pesar de la debilidad de la cadena lineal CRF para tratar con la matriz PSSM. El modelo CNF solamente con la información de la matriz PSSM obtiene 63.85 %, con la información de los aminoácidos y la matriz PSSM del experimento $Exp3$ obtiene 66.45 %. Aunque los resultados de los diferentes modelos no se pueden comparar directamente, los resultados del CNF son bastante alentadores. Teniendo en cuenta que en [53] el modelo CNF obtiene mejores resultados comparado con redes neuronales y máquinas de soporte vectorial en la predicción de estructura secundaria y reconocimiento de escritura a mano. Es posible que el modelo CNF supere la estrategia de usar dos capas independientes de clasificadores.

Las ventajas de la cadena lineal CRF y el CNF con los trabajos anteriores en la predicción de alfabetos estructurales es que explotan las relaciones estructura-estructura, modelando las relaciones entre elementos estructurales adyacentes, y explotan las relaciones secuencia-estructura con las observaciones y características externas de los aminoácidos en un sólo *framework* probabilístico.

Hasta la fecha se conoce que el trabajo con mayor Q_{16} para el alfabeto PB es el de Rangwala et. al. [56]. SvmPrat utiliza la estrategia de dos capas, su primera capa consiste de dieciséis máquinas de soporte vectorial (una máquina por cada elemento estructural). Su segunda capa, al igual que la primera consiste de dieciséis máquinas de soporte vectorial, pero reciben como entrada la salida y entrada de la primera capa. Este framework para el alfabeto PB obtiene una exactitud de 67.7 % con la información de los aminoácidos y 68.9 % agregando la información de la estructura secundaria. Aunque estos resultados no se pueden comparar directamente con los resultados de este trabajo porque se usa diferente conjunto de datos. Lo que si se puede decir es que el modelo CNF obtiene una exactitud

de 66.45 % con información de los aminoácidos y 68.53 % agregando la información de la estructura secundaria. Las ventajas del modelo CNF ante svmPrat son: primero, el modelo CNF obtiene muy buenos resultados en un sólo *framework* probabilístico mientras svmPrat usa un modelo en cascada de dos niveles que involucra 32 máquinas de soporte vectorial SVMs; segundo, el tiempo de entrenamiento del modelo CNF es mucho menor al *framework* svmPrat.

Una posible desventaja de CNF es que su capa oculta no sea la mejor manera de modelar las relaciones entre la secuencia y la estructura. Recientemente en [65] propusieron el modelo campo profundo neuronal convolucional DeepCNF (por sus siglas en ingles), que es una combinación entre un CRF y redes neuronales convolucionales poco profundas. Este modelo sobrepasa el estado del arte en predicción de estructura secundaria. Los DCNFs modelan las relaciones entre la secuencia y la estructura con una arquitectura jerárquica profunda que le permite modelar relaciones más complejas entre la entrada y la salida. Como se menciona en [65] y como trabajo futuro, los DeepCNF pueden ser considerados para predecir alfabetos estructurales.

4.5. Conclusiones

Se evalúa la capacidad del modelo CNF para predecir elementos estructurales con dos alfabetos PB y $SA_{10,3}$, y las características que mejoran el rendimiento de la predicción. El modelo CNF explota las correlaciones entre los elementos estructurales y las relaciones no lineales de la información de los aminoácidos, sin la necesidad de crear funciones características a mano. Se realizan experimentos con la siguiente información: codificación one-hot de los aminoácidos, matriz PSSM, puntajes de predicción de la estructura secundaria, y características físico-químicas. Al igual que la cadena lineal CRF, el modelo CNF con las características físico-químicas evaluadas no aportan a mejorar la predicción de ambos alfabetos. Cuando se utiliza la información de los aminoácidos se obtiene una exactitud Q_k de 52.25 % para PB y 48.65 % para $SA_{10,3}$. Agregar la información de la predicción de estructura secundaria mejora la exactitud en 13.2 % para PB y 11.21 % para $SA_{10,3}$. Usar la información de la matriz PSSM con los aminoácidos mejora la exactitud en 14.2 % para PB y 12.79 % para $SA_{10,3}$. Utilizar la información de los aminoácidos, la matriz PSSM, y el puntaje predicho de la estructura secundaria en conjunto proporciona la mejor exactitud con 68.53 % para PB y 63.29 % para $SA_{10,3}$.

Al comparar el modelo CNF con la cadena lineal CRF, el CNF lo supera con cualquier configuración de entrada evaluada en este trabajo. Con la información de los aminoácidos el CNF lo supera 9.97 % para PB y 11.34 % para $SA_{10,3}$. Con la información de aminoácidos y estructura secundaria lo supera por 2.44 % para PB y 3.79 % para $SA_{10,3}$. Con la información de aminoácidos y matriz PSSM lo supera por 8.33 % para PB y 10.04 % para $SA_{10,3}$. Con la información de los aminoácidos, estructura secundaria, y matriz PSSM lo supera por 3.9 % para PB y 4.03 % para $SA_{10,3}$.

El modelo CNF mejora los resultados del modelo CRF debido a que las transformaciones no lineales de su capa oculta crea mejores características procesadas por la cadena lineal CRF en la última capa. Estas características para el alfabeto PB ayudan a discriminar mejor

los elementos asociados a hojas y giros. Y para el alfabeto $SA_{10,3}$ ayudan a discriminar mejor los elementos relacionados con hélice-giros y hoja-giros.

El modelo CNF es bueno para modelar problemas que involucren relaciones no lineales entre la información de entrada y salida del modelo. Con la capa oculta se abstraen patrones no lineales locales, que posteriormente se combinan linealmente con los patrones de las transiciones adyacentes de las etiquetas para modelar relaciones complejas de la estructura de la proteína. A pesar de que el problema de aprendizaje deja de ser convexo, el modelo CNF logra aprender parámetros que proporcionan buenos resultados de forma eficiente.

Aunque el modelo CNF al igual que el modelo CRF logra explotar las correlaciones de los elementos estructurales adyacentes, continúa siendo un reto capturar y aprovechar las interacciones a largo rango de los aminoácidos.

Capítulo 5

Predicción de alfabetos estructurales usando una cadena lineal CRF con enlaces distantes agregados

En el problema de predicción de proteínas, todos los modelos tienen que tratar con el problema de integrar información local del contexto de los aminoácidos y la información de largo rango. Como se ha mencionado en los capítulos anteriores, integrar exitosamente la información de largo rango en los modelos es un reto. Específicamente, en los primeros trabajos de predicción de alfabetos estructurales y de estructura secundaria, se utilizó clasificadores basados en ventanas, este toma información de los aminoácidos en una ventana típicamente entre 5 y 15 aminoácidos para predecir la estructura del aminoácido central. Posteriormente, utilizaron la estrategia de dos capas independientes de clasificadores, en donde la primera capa relaciona la información local de los aminoácidos con su estructura local, y la segunda capa toma una ventana de los resultados de la primera para relacionar estructuras de las proteínas. Ambas capas recaen en la utilización de una ventana, esto significa que solo tienen en cuenta información local de las estructuras; al usar ventanas grandes el rendimiento de los modelos empeora.

Los mapas de contacto son otro tipo de predicción de estructura de proteínas, en la que es muy importante contar con modelos que puedan integrar información de largo rango entre aminoácidos. Un mapa de contacto, representa la estructura de una proteína por medio de una matriz de booleanos de dos dimensiones. Cada dimensión corresponde a las posiciones de los aminoácidos de la proteína, y un valor es verdadero, si dos aminoácidos están más cerca que una distancia límite y falso en el caso contrario. Predecir contactos de largo rango es un problema complejo, inclusive los predictores actuales tienen exactitudes muy bajas.

Un tipo de estructura común en las proteínas que involucra iteraciones a largo rango entre aminoácidos son las hojas beta. Estas crean enlaces de hidrógeno entre segmentos, de forma paralela o antiparalela. El predictor de contactos de hojas beta BETApro [14]

es una herramienta que predice contactos a partir de la estructura secundaria predicha e información evolutiva de la secuencia, tiene una precisión y sensibilidad de 40 %. Una cadena lineal CRF con saltos (skip-chain CRF) [59] es una extensión de la cadena lineal CRF que agrega enlaces entre etiquetas no adyacentes. Este tipo de modelos tiene en cuenta relaciones distantes entre las etiquetas con el fin de explotar dependencias de largo rango que ayudan a discriminar adecuadamente las etiquetas.

En este capítulo, se explora la capacidad de predicción de una cadena lineal CRF con enlaces agregados entre elementos estructurales que están en contacto. Los contactos se especifican a partir de la predicción de contactos entre residuos de hojas beta con BETApro. Se explora si agregar esta relación de largo rango o información global mejora la predicción de los elementos estructurales.

5.1. Cadena lineal CRF con enlaces agregados

La cadena lineal CRF no puede modelar dependencias distantes entre etiquetas, está limitada a relacionar etiquetas adyacentes. En el trabajo [59] extendieron la cadena lineal CRF agregando enlaces entre etiquetas distantes, como la figura 5.1. Los enlaces entre etiquetas distantes representan dependencias entre nodos distantes. Por ejemplo, en procesamiento de lenguaje natural, se pueden agregar enlaces entre las etiquetas de palabras similares con el fin de que sean etiquetadas de forma similar. Las características de los enlaces distantes pueden incorporar información del contexto de ambos nodos, de forma que si la información en un nodo es más concisa, esta pueda influenciar al otro nodo.

Este modelo necesita definir los enlaces entre etiquetas distantes. Ya que conectar cualquier par de nodos, hace que la inferencia sea intratable. Así que se necesita de una forma de definir enlaces distantes de tal forma que el grafo tenga enlaces distantes dispersos.

El modelo se define agregando otra función potencial a la cadena lineal CRF. Para una observación X, sea $I = \{(u, v)\}$ el conjunto de los pares de posiciones de secuencia que tienen enlaces distantes. La probabilidad de una secuencia de etiquetas Y dada la observación X, puede escribirse:

$$p\left(Y \mid X\right) = \frac{1}{Z\left(X\right)} \prod_{t=0}^{T-1} \psi\left(Y_t, Y_{t+1}, X, t\right) \prod_{(u,v)\epsilon I} \phi\left(Y_u, Y_v, X, u, v\right) \tag{5.1}$$

Donde Ψ es la función potencial de la cadena lineal CRF, y ϕ es la función potencial sobre los enlaces distantes. Se asume que cada función potencial se factoriza de acuerdo a unas funciones características.

$$\psi\left(Y_t, Y_{t+1}, X, t\right) = exp\left(\sum_k \theta_k f_k\left(Y_t, Y_{t+1}, X, t\right)\right) \tag{5.2}$$

$$\phi\left(Y_u, Y_v, X, u, v\right) = exp\left(\sum_k \theta_k^{'} f_k^{'}\left(Y_u, Y_v, X, u, v\right)\right) \tag{5.3}$$

Note que cada tipo de enlace tiene sus conjuntos de pesos.

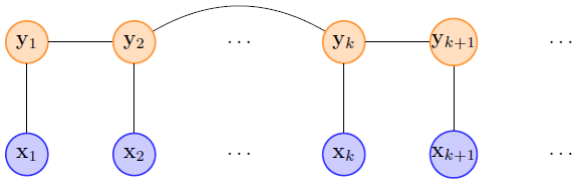


Figura 5.1: Estructura gráfica del modelo CRF con enlaces agregados.

Algunas aplicaciones de cadenas lineales con enlaces agregados

En [59] usaron una cadena lineal con enlaces agregados para extraer información. El problema consiste en etiquetar el tiempo inicial, tiempo final, lugar, y orador en un conjunto de 485 correos anunciando seminarios en la universidad Carnegie Mellon. Definieron enlaces distantes entre palabras idénticas iniciadas en mayúscula. La motivación para definir los enlaces distantes viene del aspecto más difícil en el conjunto de datos, que es identificar orador y lugar. Comparan los resultados entre la cadena lineal CRF con enlaces distantes y la cadena lineal CRF, la cadena lineal CRF con enlaces distantes tiene menos errores en etiquetar los oradores. Lo anterior ocasiona que tenga buena sensibilidad y obtenga una media F1 superior. Para la etiqueta lugar, agregar los enlaces distantes no mejora los resultados y esto se debe a que no hay mucha diferencia entre los errores de la cadena con enlaces distantes y la cadena lineal CRF.

Liu *et al.* en [45] propuso un *segmentation conditional random fields* (SCRFs) para predecir pliegues de proteínas. El modelo es una cadena lineal CRF con enlaces distantes entre etiquetas y las etiquetas son estados que representan una segmentación. Dado que los pliegues tienen una estructura 3D definida, su estructura secundaria es conservada. Se define la estructura del SCRF a partir de la estructura secundaria del pliegue. Es así como este grafo puede tener enlaces distantes, por ejemplo si hay hojas beta en el pliegue. El modelo es entrenado para reconocer el pliegue beta-helix. Este pliegue es una estructura en forma de hélice con una serie de trenzas de giros que contienen hojas beta llamados escalones. Cada sub estructura en un escalón puede tener diferente número de aminoácidos y un pliegue beta-helix puede tener diferente número de escalones, pero su estructura secundaria es conservada y por eso se puede definir un modelo gráfico con relaciones entre sus estructuras secundarias. También se compara el modelo con otros métodos como es BetaWrap y HMMER. El SCRFs es mejor reconociendo el pliegue beta-helix en diferentes familias de proteínas, ya que combina información de la secuencia y estructura en un mismo *framework*. El pliegue beta-helix es difícil de predecir porque existen interacciones largas en el pliegue, y no tiene definida una relación clara entre la estructura y la secuencia.

5.2. Especificaciones de la cadena lineal CRF con enlaces agregados

Al igual que la cadena lineal CRF, la cadena lineal CRF con enlaces agregados modela las relaciones adyacentes entre elementos estructurales. Es así como una secuencia de observaciones X corresponde a los aminoácidos $X = \{x_1, x_2, \ldots, x_n\}$, donde n es el tamaño de la proteína. La secuencia de etiquetas Y son los elementos estructurales de un alfabeto y corresponde a los elementos que codifican la proteína $Y = \{y_1, y_2, \ldots, y_m\}$, m es el número de elementos estructurales necesarios para codificarla, y $y_i \, \epsilon \{1, 2, \ldots, k\}$ donde k el número total de elementos estructurales en el alfabeto. Pero además se agregan enlaces a largo rango entre elementos estructurales asociados hojas beta. Para definir los enlaces se utiliza el predictor de contactos de hojas beta BETApro.

5.2.1. Entrenamiento y predicción

Al igual que la cadena lineal CRF el entrenamiento de la cadena lineal CRF con enlaces agregados se lleva a cabo seleccionando los parámetros que maximizan la probabilidad del conjunto de entrenamiento. En este trabajo se realiza, maximizando la log verosimilitud con la versión de memoria limitada de BFGS (algoritmo de Broyden–Fletcher–Goldfarb–Shanno). Esta técnica necesita calcular la log verosimilitud y sus derivadas, por lo que necesita hacer inferencia.

La inferencia de la cadena lineal CRF con enlaces agregados es más difícil que el de la cadena lineal CRF. Por los ciclos que pueden ser largos y sobrelapados, por eso la inferencia exacta es intratable si se tienen muchos enlaces sobrelapados. Ya que la complejidad de la inferencia requiere tiempo exponencial dado los cliques máximos del grafo. Para sobrellevar este problema se utiliza el algoritmo iterativo de propagación de creencias con ciclos, este método no garantiza que converja pero es útil en la práctica [59]. La propagación de creencias con ciclos es una generalización de los algoritmos *forward* y *backward* de HMM y las cadenas lineales CRFs.

5.2.2. Características de entrada al modelo

Las características utilizadas en la exploración al agregar enlaces distantes corresponden a la observación de los aminoácidos, la predicción de estructura secundaria, y matriz PSSM.

5.3. Métodos y materiales

El predictor de contactos de hojas beta BETApro se basa en la predicción de estructura secundaria para predecir los contactos entre hojas beta. Por esto, el predictor toma como entrada secuencias de proteínas que tienen más de una hoja beta predicha. Para conformar el conjunto de datos, se filtran las proteínas que tienen menos de una hoja beta predicha por PSIRPRE. Posteriormente, se predicen los contactos entre residuos asociados a hojas beta

	a	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**j**	**k**	**l**	**m**	**n**	**o**	**p**
a	2	2	8	91	5	23	1	6	1	0	4	1	1	0	0	0
b	-	4	30	100	8	18	0	6	3	0	1	0	2	0	0	2
c	-	-	88	790	81	76	5	18	0	3	3	3	6	0	1	1
d	-	-	-	4818	145	200	3	37	10	7	12	10	16	2	0	9
e	-	-	-	-	4	6	0	1	1	0	2	1	0	0	0	3
f	-	-	-	-	-	22	0	0	1	0	3	2	2	0	0	2
g	-	-	-	-	-	-	0	0	0	0	0	0	1	0	0	0
h	-	-	-	-	-	-	-	0	0	0	0	2	1	0	0	1
I	-	-	-	-	-	-	-	-	0	0	0	0	1	0	0	0
j	-	-	-	-	-	-	-	-	-	4	0	0	0	0	0	0
k	-	-	-	-	-	-	-	-	-	-	2	0	0	0	0	0
l	-	-	-	-	-	-	-	-	-	-	-	0	0	0	1	0
m	-	-	-	-	-	-	-	-	-	-	-	-	6	0	0	1
n	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0	0
o	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0	0
p	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	0

Tabla 5.1: Frecuencia de los elementos estructurales de los contactos de hojas beta predichos del conjunto de datos.

con BETApro. Se definen los enlaces a largo rango de acuerdo a los contactos predichos con un umbral de predicción mayor o igual a 0.7 y distancia entre contactos mayor a 6 aminoácidos. Se conforma un conjunto de datos con 735 proteínas para entrenar y con 175 para probar.

En esta exploración se utiliza el alfabeto estructural PB. Se compara las predicciones de una cadena lineal CRF con enlaces agregados y una cadena lineal CRF. Para evaluar la capacidad de mejora en la predicción al agregar los enlaces distantes entre elementos asociados a hojas beta.

5.4. Resultados y análisis

Frecuencia de los elementos estructurales en los contactos

La tabla 5.1 contiene la frecuencia de los elementos estructurales de los contactos de hojas beta predichos del conjunto de datos. Los elementos más frecuentes son los asociados a hojas beta y giros, como son a, b, c, e, y f según el análisis con la asignación de estructura secundaria usando DSSP del capítulo 3. El elemento más frecuente es d, que es exclusivo de hojas beta. De esta tabla de frecuencias, se espera que al entrenar una cadena lineal CRF con contactos, este aprenda de las relaciones más frecuentes entre elementos estructurales.

Predicción con una cadena lineal CRF con enlaces agregados

Se entrena una cadena lineal CRF con los contactos predichos y una cadena lineal CRF con las mismas funciones características. La tabla 5.2 contiene los resultados Q_{16} de cada modelo. La cadena lineal CRF con los contactos predichos obtiene 54.62 % y la cadena lineal CRF obtiene 54.74 %. La cadena lineal CRF sin contactos obtiene mejor exactitud Q_{16}. Para analizar porque sucede esto, se calcula las métricas de precisión y sensibilidad a los elementos predichos en los contactos para ambos modelos. La tabla 5.3 contiene los resultados, esta indica que la cadena lineal CRF predice bien los elementos estructurales donde se relacionan contactos de hojas beta. Por esta razón, al agregarlos en la cadena lineal CRF con enlaces agregados no se obtienen buenos resultados.

Modelo/Alfabeto	**Predicción PB Q_{16}**
CRF con enlaces agregados	54.62
CRF	54.74

Tabla 5.2: Resultados de la exactitud Q_{16} del alfabetos PB para los modelos CRF con enlaces agregados y CRF.

	CRF con contactos		**CRF**	
Etiqueta	**Precisión**	**Sensibilidad**	**Precisión**	**Sensibilidad**
a	0.3666	0.3793	0.45	0.3103
b	0.0	0.0	0.25	0.0357
c	0.6207	0.1905	0.5494	0.2645
d	0.7801	0.9561	0.8038	0.9404
e	0.5217	0.2727	0.5384	0.4772
f	0.3333	0.0625	0.4285	0.1875
g	0.0	0.0	0.0	0.0
h	0.3846	0.4166	0.4615	0.5
i	0.0	0.0	0.0	0.0
j	0.0	0.0	1.0	0.5
k	0.0	0.0	0.0	0.0
l	0.0	0.0	0.0	0.0
m	0.0	0.0	0.0	0.0
n	0.0	0.0	0.0	0.0
o	0.0	0.0	0.0	0.0
p	0.0	0.0	0.5	0.6666

Tabla 5.3: Precisión y sensibilidad para cada elemento estructural del alfabeto PB en solo los contactos.

5.5. Conclusiones

En este capítulo se explora si agregar contactos predichos entre residuos de hojas beta por BETApro en una cadena lineal CRF, mejora la exactitud de las predicciones de elementos

estructurales. Al entrenar la cadena lineal CRF con contactos no se mejora los resultados de predicción comparada con una cadena lineal CRF. Se cree que el CRF con contactos no mejora los resultados, debido a que no hay relaciones útiles que aprender de los contactos, ya que la cadena lineal CRF sin estos puede predecir bien los elementos de hojas beta.

Por otra parte, entrenar una cadena lineal CRF con enlaces es más difícil que una cadena lineal CRF, si se tienen muchos enlaces es posible que el modelo no converja. Es por esto, que es muy complejo modelar las relaciones a largo rango de proteínas con modelos gráficos no dirigidos. Si se tienen en cuenta en un modelo, hace que este sea computacionalmente intratable.

Capítulo 6

Comparación de algunos modelos en la predicción del alfabeto PB

A continuación, se presentan los resultados de cuatro modelos evaluados en la predicción del alfabeto estructural PB en un pequeño conjunto de datos. Los modelos usados son: una implementación de la red neuronal definida en [23], el *framework* SvmPrat [56], un campo aleatorio condicional lineal CRF, y un campo neuronal condicional CNF.

6.1. Materiales y métodos

Todos los modelos reciben como entrada la misma información que corresponde a la matriz PSSM y el puntaje de la estructura secundaria predicha por PSSPRED [68]. El conjunto de datos es de 480 secuencias y estructuras de proteínas no homólogas seleccionadas aleatoriamente del conjunto de datos de 3052 secuencias definidas en el capítulo 3 sección 3.2. Los resultados de los experimentos se obtienen por validación cruzada de tres iteraciones sobre el conjunto de datos; 2/3 de los datos se utilizan para entrenar y 1/3 para de los datos para validar. Al igual que los capítulos anteriores, la evaluación de los resultados se lleva a cabo con la exactitud Q_{16}, precisión, sensibilidad, y coeficiente de correlación de Matthews MCC. Para información detallada dirigirse al capítulo 3 sección 3.2.

6.1.1. Descripción de los modelos

- **Red Neuronal:** La red neuronal tiene a la misma arquitectura de dos capas descrita en el trabajo de Dong et. al. [23]. Es un modelo de dos capas, donde cada capa es una red neuronal con una capa oculta. La primera red se encarga de abstraer los patrones secuencia-estructura y la segunda recibe los resultados de la primera red y los usa para abstraer relaciones estructura-estructura. A diferencia de la red de Dong, la implementación en este trabajo recibe además de la matriz PSSM, los puntajes predichos de la estructura secundaria. Y utiliza la función *tanh* como activación de las neuronas ocultas. Por lo anterior se realizan experimentos variando el número de neuronas y la ventana de contexto con los que se obtiene un buen rendimiento

en la predicción. Los valores de los parámetros seleccionados con los que se obtiene el mejor rendimiento son: para la primera red neuronal una ventana de tamaño 9 y 100 unidades ocultas; para la segunda red una ventana de tamaño 5 y 100 unidades ocultas.

- **SvmPrat:** es un *framework* que utiliza un modelo en cascada de dos capas con máquinas de soporte vectorial SVM. Cada capa del modelo en cascada tiene K uno versus el resto de clasificadores binarios. Esto quiere decir que para el problema de predicción de alfabetos con PB tiene 16 clasificadores binarios por capa, en total son 32. La segunda capa del modelo recibe los resultados de la capa anterior, más la información de entrada de la primera capa. SvmPrat proporciona dos funciones kernel, la utilizada en este estudio corresponde a un kernel de segundo orden. La ventana de contexto utilizada en el entrenamiento es de tamaño 19 (para tener una ventana de tamaño 19 los parámetros w y f toman los valores $w = 9$ y $f = 9$) [56].
- **Cadena lineal CRF:** El modelo campo aleatorio condicional lineal fue descrito en el capítulo 3, es un modelo probabilístico que combina la funciones características de las observaciones y transiciones entre elementos estructurales adyacentes linealmente. El parámetro de regularización y ventana seleccionados para la comparación son: tamaño de ventana 5 y parámetro de regularización 9.
- **Cadena CNF:** El modelo campo neuronal condicional es un modelo probabilístico que agrega una capa oculta que hace transformaciones no lineales a las observaciones y luego estas se combinan linealmente con las transiciones de las etiquetas. Este modelo se describe en el capítulo 4, los parámetros seleccionados para la comparación son: ventana tamaño 9, 60 neuronas ocultas, y parámetro de regularización igual a 9.

6.2. Resultados y análisis

La tabla 6.1 muestra los resultados de los modelos en el pequeño conjunto de datos usado. El rendimiento de todos los modelos se ve muy afectado por el tamaño del conjunto de datos, pero debido a que se contaba con un límite de recursos computacionales y a la gran cantidad de tiempo de entrenamiento requerido por SvmPrat, se realiza la comparación entre los modelos con el pequeño conjunto de datos.

Al entrenar la red neuronal con los puntajes de la estructura secundaria como entrada, la segunda capa solo mejora en 1 % en promedio los resultados de la capa anterior. En [23] la utilización de la segunda capa mejora los resultados en 2 %.

El modelo CNF tiene la mejor exactitud de predicción, el segundo es la red neuronal, seguido por SvmPrat, y de último el modelo CRF. Se aprecia que los modelos que hacen transformaciones no lineales a la información de entrada presentan mejores resultados, a pesar que el modelo CRF modela secuencialmente los elementos estructurales. Además la estrategia de los modelos SvmPrat y la red neuronal de usar una segunda capa para modelar las relaciones estructura-estructura no es mejor que el modelo CNF que modela secuencialmente las relaciones adyacentes entre los elementos, y de forma conjunta la relación no lineal secuencia-estructura.

Modelo	Q_{16}
Red Neuronal (RN)	64.27
SvmPrat	63.62
CRF	61.77
CNF	64.44

Tabla 6.1: Resultados de la exactitud Q_{16} del alfabeto PB para diferentes modelos evaluados.

La tabla 6.2 muestra la precisión, sensibilidad, y MCC en los modelos evaluados. Todos los modelos tienen los mejores resultados con los elementos m y d, asociados a hélices y hojas, respectivamente. Los modelos tienen resultados malos para los elementos g, j, h, i, y b. Los elementos j e i están asociados a giros, los elementos b y h son frecuentes en hojas-giros, y el elemento g es frecuente en los tres estados de estructura secundaria. Además, los modelos tienen resultados regulares para los elementos c, e, y f. Estos elementos están asociados a hojas-giros.

Al comparar los resultados MCC de SvmPrat con los otros modelos, el elemento g tiene el peor resultado y es superado ampliamente por los demás. También, los resultados MCC de los elementos h, e, y o son superados con una gran diferencia. Para el modelo CRF, los resultados MCC de los elementos c, k, y l son superados ampliamente por los demás. Los resultados MCC del elemento b de los modelos SvmPrat y CRF son superados ampliamente por los demás. Los modelos RN y CNF superan a los resultados de los demás en casi todos los elementos.

6.3. Conclusiones

A pesar de llevar a cabo esta comparación con un pequeño conjunto de datos, el modelo CNF logra obtener mejores resultados que los otros modelos. El framework SvmPrat y la red neuronal sobrepasan en exactitud a la cadena lineal CRF. Al detallar los resultados por elemento estructural todos los modelos se les facilita predecir los elementos asociados a hélices y el elemento d asociado a hojas beta. Los elementos asociados a hojas y giros son más difíciles de predecir.

Modelar la proteína en un modelo gráfico no dirigido como una cadena en la que se explotan las relaciones entre elementos estructurales adyacentes y las relaciones no lineales entre la secuencia y los elementos, permite obtener los mejores resultados con el CNF. Lo que indica que es una alternativa interesante y prometedora contrastada con la metodología de dividir el problema en capas independientes utilizadas en svmPrat y la red neuronal.

		Precisión				Sensibilidad				MCC			
Etiqueta	**Frecuencia**	**CRF**	**SvmPrat**	**RN**	**CNF**	**CRF**	**SvmPrat**	**RN**	**CNF**	**CRF**	**SvmPrat**	**RN**	**CNF**
a	3.82	51.32	45.42	55.56	51.91	59.15	57.16	51.79	53.3	53.25	48.84	51.91	50.74
b	4.46	35.32	44.78	40.61	40.15	17.34	13.85	23.05	21.39	22.41	23.15	28.3	27.06
c	8.52	46.77	45.64	47.75	47.44	33.65	43.45	44.66	44.4	35.27	39.86	41.65	41.38
d	18.9	62.16	60.88	61.46	61.84	73.65	79.52	77.91	78.23	60.43	62.07	61.66	62.17
e	2.18	42.23	41.05	45.81	43.64	35.94	31.95	36.21	34.97	37.72	34.98	39.55	37.86
f	6.61	47.57	47.68	47.67	50.36	40.37	43.53	46.69	44.3	40.35	42.07	43.64	43.93
g	1.15	21.87	26.02	30.52	39.6	6.44	1.4	7.39	5.22	11.35	5.79	14.56	14.04
h	2.18	39.13	41.15	43.89	40.31	36.11	27.11	30.42	34.22	36.3	32.23	35.41	35.89
i	1.56	34.28	39.76	44.49	41.89	39.45	32.4	34.95	34.27	35.73	34.98	38.57	37.02
j	0.88	31.19	35.96	25.86	28.15	7.92	5.43	10.55	8.18	15.27	13.69	16	14.79
k	5.41	52.39	49.65	54.33	54.78	47.25	57.09	53.08	53.31	47.12	50.45	51.16	51.52
l	5.25	53.68	56.35	56.4	57.72	43.61	47.85	47.23	47.38	45.84	49.52	49.18	49.95
m	31.51	77.95	85.2	81.22	80.59	89.25	86.29	88.43	89.14	75.69	79.26	77.41	77.34
n	1.79	53.47	61.61	65.55	66.95	59.42	52.85	52.27	51.99	55.51	56.31	57.84	58.33
o	2.47	56.47	48.43	60.94	59.71	54.49	58.28	53.05	51.93	54.37	51.84	55.82	54.65
p	3.31	45.74	46.43	53.81	53.4	45.48	47.18	43.7	44.04	43.76	44.97	46.86	46.9

Tabla 6.2: Precisión, sensibilidad, y MCC para cada elemento estructural del alfabeto PB en los modelos evaluados.

Capítulo 7

Conclusiones y trabajo futuro

En este trabajo de investigación, se explora el modelo Campo Aleatorio Condicional CRF para predecir los alfabetos estructurales PB y $SA_{10,3}$. La estructura nativa de la proteína se codificó de forma local con los alfabetos. Por esto, la predicción de estructura de proteína con alfabetos estructurales es una aproximación del *backbone* de la estructura nativa de la proteína.

Basados en la información de la secuencia se aplicaron diferentes modelos CRFs secuenciales que capturan las interacciones entre los elementos estructurales y las relaciones entre los elementos y la información de la secuencia. Mostrando la eficacia de los campos aleatorios condicionales para predecir estructura local a partir de los alfabetos estructurales. Está exploración comprueba de forma empírica que los campos aleatorios condicionales son útiles para predecir elementos estructurales.

7.1. Conclusiones

- Se conforma una base de datos de 3052 proteínas, representativa del PDB para entrenar y evaluar los modelos CRFs.
- Se realizó un análisis de frecuencias de los alfabetos en los estados asignados por DSSP en el conjunto de datos, para caracterizar los elementos estructurales. El alfabeto PB tiene elementos estructurales exclusivos para cada estado de estructura secundaria. Pero este alfabeto tiene elementos que son frecuentes en el estado hojas beta y giros. El alfabeto $SA_{10,3}$ también presenta elementos exclusivos para cada estado de estructura secundaria. Pero tiene elementos frecuentes en hélices-giros y hojas-giros.
- Se usó una cadena lineal CRF para predecir los alfabetos estructurales. Esta permite explotar las correlaciones de los elementos estructurales vecinos e integrar la información local de las observaciones de aminoácidos. La mejor exactitud de 64.63 % para PB y 59.03 % para $SA_{10,3}$ se obtiene con la información de los aminoácidos, la estructura secundaria predicha, y la matriz PSSM. La cadena lineal CRF presenta

una debilidad al tratar con la información de la matriz PSSM, ya que este modelo hace una combinación lineal y con esta matriz se obtiene buenos resultados en modelos que hacen transformaciones no lineales.

- Se usó una Cadena Neuronal Condicional CNF para predecir los alfabetos estructurales. Al igual que en la cadena lineal CRF, se explota las correlaciones entre los elementos estructurales vecinos. Pero a diferencia de la cadena CRF, la capa neuronal con las transformaciones no lineales ayuda a aprender mejores relaciones entre la estructura local y la información de la secuencia sin necesidad de crear funciones características a mano. La mejor exactitud del CNF se obtiene con las mismas características que la cadena lineal CRF, 68.53 % para PB y 63.29 % para $SA_{10,3}$.

- El uso de funciones características físico-químicas de los aminoácidos en ambos modelos no mejora el rendimiento de la predicción.

- Se compararon la cadena lineal CRF y el modelo CNF. El CNF al hacer transformaciones no lineales con la capa neuronal sobrepasa a la cadena lineal CRF. La información de la estructura secundaria predicha ayuda a mejorar notablemente el rendimiento de la predicción en ambos modelos. El CNF al hacer transformaciones no lineales, mejora considerablemente los resultados de la cadena lineal CRF cuando se utiliza la observaciones de los aminoácidos y/o la matriz PSSM.

- Se contrasto los resultados de predicción de los elementos de ambos modelos con la caracterización según la frecuencia en estados de estructura secundaria. Los elementos del alfabeto PB frecuentes en hojas beta y giros y los elementos del alfabeto $SA_{10,3}$ frecuentes en hojas beta-giros y hélices-giros son los más difíciles de predecir.

- Se realizó una comparación con un pequeño conjunto de datos entre los modelos CNF, CRF, una red neuronal, y el *framework* SvmPrat. La red neuronal y SvmPrat utilizan la estrategia de dos capas independientes de clasificadores. El modelo CNF supera en exactitud a los demás modelos, lo que indica que es una alternativa interesante y prometedora contrastada con la metodología de dos capas independientes.

- No se puede comparar directamente los resultados del CNF con otros modelos porque utilizan diferentes conjuntos de datos de entrenamiento. Sin embargo, la exactitud de los diferentes modelos para su conjunto de datos especifico puede dar una idea de la bondad de cada uno. El CNF para el alfabeto PB en una validación cruzada de cinco iteraciones obtiene 68.53 %. El *framework* SvmPrat (estado del arte) para PB en una validación cruzada de tres iteraciones obtiene 68.9 %. Se observa que sus resultados son cercanos, aunque no se puede decir cuál es mejor, el CNF tiene las ventajas de ser un solo modelo probabilístico que se puede entrenar más rápido que SvmPrat.

- La exactitud de la predicción de los modelos CRF y CNF se ve afectada si una proteína tiene muy pocos homólogos, ya que la matriz PSSM seria poco informativa y la predicción de estructura secundaria también depende de esta. Pero a pesar de esto, el modelo CNF obtiene 52.25 % para PB y 48.65 % para $SA_{10,3}$ solamente con la observación de los aminoácidos.

- Se usó una cadena lineal CRF con enlaces agregados para predecir el alfabeto PB. Se utilizó un predictor de contactos de residuos en hojas beta, llamado BETApro

para definir los enlaces agregados a la cadena lineal CRF. Lo anterior se hace con la hipótesis de que al agregar los enlaces, el modelo aprende relaciones de largo rango entre los elementos estructurales que mejoran la exactitud de la predicción. Pero al entrenar el modelo, este no mejora la exactitud al compararlo con una cadena lineal CRF. Lo anterior no se debe a que no hay relaciones entre las etiquetas enlazadas, sino a que la cadena lineal CRF es capaz de predecir bien los elementos donde se agregaron los enlaces.

- Aunque las cadenas aleatorias condicionales CRFs implementadas en este trabajo, aprovechan las correlaciones entre los elementos vecinos en la secuencia de una proteína y explota las relaciones entre los elementos y la secuencia, solo se aprovechan las relaciones cercanas. Al tratar de incluir relaciones de largo rango con los enlaces agregados, no se mejoraron los resultados, pero se incrementó la complejidad del modelo, tanto que este puede no converger.

7.2. Trabajo futuro

- Basado en los buenos resultados del modelo CNF, la debilidad de la cadena lineal CRF de tratar con información como la matriz PSSM, y por los actuales avances de predicción de estructura secundaria con *deep learning*, se recomienda en trabajos futuros explorar modelos de predicción de alfabetos estructurales que exploten relaciones a corto y largo rango de los aminoácidos, por medio de jerarquías a partir de la información de estos. Este tipo de modelos basados en *deep learning* aprenden relaciones por medio de transformaciones no lineales con arquitecturas de redes neuronales que son difíciles de aprender con modelos lineales. Además, se recomienda explorar y aplicar los avances en los métodos de entrenamiento de estas redes que mejoran la exactitud de los modelos.

- En este trabajo de investigación se mostró que los CRFs explotan la fuerte correlación entre elementos estructurales adyacentes; lo que produce buenos resultados en exactitud. Pero se sabe, que en los problemas de predicción de secuencias es muy importante el contexto. En la predicción de estructura de proteínas, se necesita contar con métodos que integren la información del contexto a corto y largo rango de la interacción de los aminoácidos. Por este motivo se recomienda explorar modelos de los avances en la predicción de secuencia a secuencia. Este tipo de modelos tratan de tener a la mano información del contexto para hacer predicciones locales. Este problema no es único en la predicción de estructura de proteínas y se presenta en diversos campos como procesamiento del lenguaje natural, reconocimiento de voz y visión por computador.

- Dado que existen múltiples problemas en proteómica donde la información estructural es muy útil, se recomienda usar los resultados de la predicción de alfabetos estructurales en: *threading* o ensamble de fragmentos, alineamientos estructurales, predicción de clase, clasificación de *folds*, y predicción de superfamilias.

Referencias Bibliográficas

[1] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, Peter Walter, et al. *Molecular Biology of the Cell*, chapter Proteins. Garland Science, 2002.

[2] Pierre Baldi, Søren Brunak, Paolo Frasconi, Giovanni Soda, and Gianluca Pollastri. Exploiting the past and the future in protein secondary structure prediction. *Bioinformatics*, 15(11):937–946, 1999.

[3] Cristina Benros, Alexandre G. de Brevern, Catherine Etchebest, and Serge Hazout. Assessing a novel approach for predicting local 3d protein structures from sequence. *Proteins: Structure, Function, and Bioinformatics*, 62:865–880, 2005.

[4] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov, and Philip E. Bourne. The protein data bank. *Nucleic Acids Research*, 28, 2000.

[5] Axel Bernal, Koby Crammer, Artemis Hatzigeorgiou, and Fernando Pereira. Global discriminative learning for higher-accuracy computational gene prediction. *PLoS Comput Biol*, 3(3):e54, 2007.

[6] Christopher M Bishop. *Pattern recognition and machine learning*, volume 1. springer New York, 2006.

[7] Aurélie Bornot, Catherine Etchebest, and Alexandre G De Brevern. A new prediction strategy for long local protein structures using an original description. *Proteins: Structure, Function, and Bioinformatics*, 76(3):570–587, 2009.

[8] Christopher Bystroff and David Baker. Prediction of local structure in proteins using a library of sequence-structure motifs. *Journal of Molecular Biology*, 281:565–577, 1998.

[9] Christopher Bystroff, Vesteinn Thorsson, and David Baker. Hmmstr: a hidden markov model for local sequence-structure correlations in proteins. *Journal of Molecular Biology*, 301:173–190, 2000.

[10] A. C Camproux, R Gautier, and P Tuffer. A hidden markov model derived structural alphabet for proteins. *Journal of Molecular Biology*, 339:591–605, 2004.

[11] A. C Camproux, P Tuffer, J. P Chevrolat, J. F Boisvieux, and S. Hazout. Hidden markov model approach for identifying the modular framework of the protein backbone. *Protein Engineering*, 12(12):1063–1073, 1999.

[12] Ching-Tai Chen, Hsin-Nan Lin, Ting-Yi Sung, and Wen-Lian Hsu. Hyplosp: a knowledge-based approach to protein local structure prediction. *Journal of bioinformatics and computational biology*, 4(06):1287–1307, 2006.

[13] Gang Chen et al. Sequential labeling with online deep learning. *arXiv preprint arXiv:1412.3397*, 2014.

[14] Jianlin Cheng and Pierre Baldi. Three-stage prediction of protein β-sheets by neural networks, alignments and graph algorithms. *Bioinformatics*, 21(suppl 1):i75–i84, 2005.

[15] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[16] Alexandre De Brevern, Cristina Benros, and Serge Hazout. Structural alphabet: from a local point of view to a global description of protein 3d structures. *Bioinformatics: new research*, pages 127–69, 2005.

[17] Alexandre G de Brevern. New assessment of a structural alphabet. *In silico biology*, 5(3):283–289, 2005.

[18] Alexandre G. de Brevern, Catherine Etchebest, Cristina Benros, and Serge Hazout. "pinning strategy": a novel approach for predicting the backbone structure in terms of protein blocks from sequence. *J. Biosci*, 19(10):51–70, 2007.

[19] Alexandre G. de Brevern, Catherine Etchebest, and Serge Hazout. Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins: Structure, Function, and Genetics*, 41:271–287, 2000.

[20] Alexandre G. de Brevern, H. Valadie, Serge Hazout, and Catherine Etchebest. Extension of a local backbone description using a structural alphabet: a new approach to the sequence-structure relationship. *Bioinformatics*, 19(10):2871–2886, 2002.

[21] Trinh Do, Thierry Arti, et al. Neural conditional random fields. In *International Conference on Artificial Intelligence and Statistics*, pages 177–184, 2010.

[22] Qiwen Dong, Xiaolong Wang, and Lei Lin. Prediction of protein local structures and folding fragments based on building-block library. *Proteins: Structure, Function, and Bioinformatics*, 72:353–366, 2008.

[23] Qiwen Dong, Xiaolong Wang, Lei Lin, and Yadong Wang. Analysis and prediction of protein local structure based on structure alphabets. *Proteins: Structure, Function, and Bioinformatics*, 72:163–172, 2008.

[24] Catherine Etchebest, Cristina Benros, Serge Hazout, and Alexandre G. de Brevern. A structural alphabet for local protein structures: Improved prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 59:810–827, 2005.

[25] Laurent Fourrier, Cristina Benros, and Alexandre G De Brevern. Use of a structural alphabet for analysis of short loops connecting repetitive structures. *BMC bioinformatics*, 5(1):58, 2004.

[26] Glennie Helles. A comparative study of the reported performance of ab initio protein structure prediction algorithms. *Journal of the Royal Society Interface*, 5(21):387–396, 2008.

[27] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[28] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[29] David T Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202, 1999.

[30] Agnel Praveen Joseph, Garima Agarwal, Swapnil Mahajan, Jean-Christophe Gelly, Lakshmipuram S Swapna, Bernard Offmann, Frédéric Cadet, Aurélie Bornot, Manoj Tyagi, Hélène Valadié, et al. A short survey on protein blocks. *Biophysical Reviews*, 2(3):137–145, 2010.

[31] Agnel Praveen Joseph, Aurelie Bornot, and Alexandre G. De Brevern. *Local Structure Alphabets*. Wiley I& sons,Inc, 2010.

[32] Agnel Praveen Joseph and Alexandre G. de Brevern. From local structure to a global framework: recognition of protein folds. *J. R. Soc. Interface*, 11(95):175–187, 2014.

[33] Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

[34] Rachel Karchin, Melissa Cline, Yael Mandel-Gutfreund, and Kevin Karplus. Hidden markov models that use predicted local structure for fold recognition: alphabets of backbone geometry. *Proteins: Structure, Function, and Bioinformatics*, 51(4):504–514, 2003.

[35] Hyunsoo Kim and Haesun Park. Protein secondary structure prediction based on an improved support vector machines approach. *Protein Engineering*, 16(8):553–560, 2003.

[36] Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[37] Rachel Kolodny, Patrice Koehl, Leonidas Guibas, and Michael Levitt. Small libraries of protein fragments model native protein structures accurately. *Journal of Molecular Biology*, 323:297–307, 2002.

[38] John D. Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[39] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[40] Jooyoung Lee, Sitao Wu, and Yang Zhang. Ab initio protein structure prediction. In *From protein structure to function with bioinformatics*, pages 3–25. Springer, 2009.

[41] Jooyoung Lee, Sitao Wu, and Yang Zhang. *Ab InitioProtein Structure Prediction*. Springer Science + Business Media B.V., 2009.

[42] Quan Li, Changhai Zhou, and Haiyan Liu. Fragment-based local statistical potentials derived by combining an alphabet of protein local structures with secondary structures and solvent accessibilities. *Proteins: Structure, Function, and Bioinformatics*, 74:820–836, 2009.

[43] Yan Liu. *Conditional graphical models for protein structure prediction*. PhD thesis, University of Pittsburgh, 2006.

[44] Yan Liu, Jaime Carbonell, Judith Klein-Seetharaman, and Vanathi Gopalakrishnan. Comparison of probabilistic combination methods for protein secondary structure prediction. *Bioinformatics*, 20(17):3099–3107, 2004.

[45] Yan Liu, Jaime Carbonell, Peter Weigele, and Vanathi Gopalakrishnan. Protein fold recognition using segmentation conditional random fields (scrfs). *Journal of Computational Biology*, 13(2):394–406, 2006.

[46] Lior Lukov, Sanjay Chawla, Wei Liu, Brett Church, and Gaurav Pandey. *Protein Secondary Structure Prediction with Conditional Random Fields*. School of Information Technologies, University of Sydney, 2010.

[47] Jianzhu Ma, Jian Peng, Sheng Wang, and Jinbo Xu. A conditional neural fields model for protein threading. *Bioinformatics*, 28(12):i59–i66, 2012.

[48] Jianzhu Ma and Sheng Wang. Acconpred: Predicting solvent accessibility and contact number simultaneously by a multitask learning framework under the conditional neural fields model. *BioMed Research International*, 2015, 2015.

[49] Andrew McCallum, Dayne Freitag, and Fernando Pereira. Maximum entropy markov models for information extraction and segmentation. *Proc. ICML*, pages 591–598, 2000.

[50] Jens Meiler, Michael Müller, Anita Zeidler, and Felix Schmäschke. Generation and evaluation of dimension-reduced amino acid parameter representations by artificial neural networks. *Molecular modeling annual*, 7(9):360–369, 2001.

[51] Kevin Patrick Murphy. *Undirected graphical models (Markov random fields)*. The MIT Press, 2012.

[52] Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007.

[53] Jian Peng, Liefeng Bo, and Jinbo Xu. Conditional neural fields. In *Advances in neural information processing systems*, pages 1419–1427, 2009.

[54] Rohit Prabhavalkar and Eric Fosler-Lussier. Backpropagation training for multilayer conditional random field based phone recognition. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5534–5537. IEEE, 2010.

[55] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[56] Huzefa Rangwala, Christopher Kauffman, and George Karypis. svmprat: Svm-based protein residue annotation toolkit. *BMC Bioinformatics*, 10(1):439, 2009.

[57] Oliver Sander, Ingolf Sommer, and Thomas Lengauer. Local protein structure prediction using discriminative models. *BMC bioinformatics*, 7(1):14, 2006.

[58] Sunita Sarawagi and William W. Cohen. Semi-markov conditional random fields for information extraction. *Advances in Neural Information Processing Systems*, pages 1185–1192, 2004.

[59] Charles Sutton and Andrew McCallum. Collective segmentation and labeling of distant entities in information extraction. Technical report, DTIC Document, 2004.

[60] Charles Sutton and Andrew McCallum. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2011.

[61] Charles Sutton, Khashayar Rohanimanesh, and Andrew McCallum. Dynamic conditional random fields: Factorized probabilistic models for labeling and segmenting sequence data. *The Journal of Machine Learning Research*, 8:693–723, 2007.

[62] Nilson Mossos Vivas. *Predicción estructural de una proteína basada en elementos estructurales tridimensionales y en propiedades fisicoquímicas*. PhD thesis, Escuela de ingeniería de sistemas y computación, Universidad del Valle, 2016.

[63] Guoli Wang and Roland L. Dunbrack. Pisces: a protein sequence culling server. *Bioinformatics*, 19(12):1589–1591, 2003.

[64] Mengqiu Wang and Christopher D Manning. Effect of non-linear deep architecture in sequence labeling. In *Proceedings of the 6th International Joint Conference on Natural Language Processing (IJCNLP)*, 2013.

[65] Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific reports*, 6, 2016.

[66] Sy Bor Wang, Ariadna Quattoni, Louis-Philippe Morency, David Demirdjian, and Trevor Darrell. Hidden conditional random fields for gesture recognition. *Computer Vision and Pattern Recognition*, 2, 2006.

[67] Zhiyong Wang, Feng Zhao, Jian Peng, and Jinbo Xu. Protein 8-class secondary structure prediction using conditional neural fields. *Proteomics*, 11(19):3786–3792, 2011.

[68] Renxiang Yan, Dong Xu, Jianyi Yang, Sara Walker, and Yang Zhang. A comparative assessment and analysis of 20 representative sequence alignment methods for protein structure prediction. *Scientific reports*, 3, 2013.

[69] An-Suei Yang and Lu yong Wang. Local structure prediction with local structure-based sequence profiles. *Bioinformatics*, 19(10):1267–1274, 2003.

[70] Dong Yu, Shizhen Wang, and Li Deng. Sequential labeling using deep-structured conditional random fields. *Selected Topics in Signal Processing*, 4(6), 2010.

[71] Xin Yuan and Christopher Bystroff. Protein contact map prediction. In Ying Xu, Dong Xu, and Jie Liang, editors, *Computational Methods for Protein Structure Prediction and Modeling*, Biological and Medical Physics, Biomedical Engineering, pages 255–277. Springer New York, 2007.

[72] Shesheng Zhang, Shengping Jin, and Bin Xue. Accurate prediction of protein dihedral angles through conditional random field. *Frontiers in Biology*, 8(3):353–361, 2013.

[73] Yang Zhang. Interplay of i-tasser and quark for template-based and ab initio protein structure prediction in casp10. *Proteins: Structure, Function, and Genetics*, 82:175–187, 2014.

[74] Feng Zhao, Jian Peng, and Jinbo Xu. Fragment-free approach to protein folding using conditional neural fields. *Bioinformatics*, 26(12):i310–i317, 2010.

[75] Olav Zimmermann and Ulrich H. E. Hansmann. Locustra: Accurate prediction of local protein structure using a two-layer support vector machine approach. *Journal of Chemical Information and Modeling*, 48(9):1903–1908, 2008.

Apéndice A

Conjunto de datos

A continuación se presenta la descripción del conjunto de datos usado para entrenar y probar los modelos del capítulo 3 y 4. El conjunto de datos será anexado a este trabajo en un archivo de texto con el propósito de que se use en futuros trabajos, el archivo tiene siguiente formato:

Cada proteína comienza en una nueva línea con el símbolo '>', seguido del nombre de la proteína, posteriormente cada aminoácido corresponde a una nueva línea con la siguiente información, separada por TABs:

- Símbolo del aminoácido: 'A', 'C', 'E', 'D', 'G', 'F', 'I', 'H', 'K', 'M', 'L', 'N', 'Q', 'P', 'S', 'R', 'T', 'W', 'V', 'Y'.
- Etiqueta asignada del alfabeto PB: 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'Z'.
- Etiqueta asignada del alfabeto $SA_{10,3}$: 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'Z'.
- Información de la matriz PSSM correspondiente al aminoácido con el orden: A, R, N, D, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V.
- Puntaje predicho de la estructura secundaria con el orden 'H', 'E', 'C'.
- Asignación DSSP: 'H','B','E','G','I','T','S','-'.
- Asignación de accesibilidad relativa, la medida se obtiene por medio del programa DSSP y se crea un alfabeto de tres estados según sus valores: <9 %: interno, 9–36 %: intermedio; >36 %: expuesto: 'B', 'I', 'E'.

Cada línea contiene 28 ítems, así la proteína es descrita por una matriz de $Nx28$ ítems, donde N es el tamaño de la proteína. Por ejemplo:

>1pl8a
A Z Z 4 -3 -2 -3 -2 -2 -2 -2 -3 -4 -4 -2 -3 -4 -3 5 -1 -5 -4 -3 0.959 0.022 0.012 C E
A Z h 3 -2 -1 -2 -3 -1 0 -2 0 -3 -3 -1 -3 -2 4 1 2 0 -2 -2 0.925 0.029 0.036 C B
A d h 2 -2 -1 -2 -3 0 0 -2 -2 -3 -3 -1 -3 -4 3 2 3 -4 -3 0 0.920 0.033 0.044 C B
A d h 2 0 0 0 -2 -1 0 -1 -2 0 -1 -1 0 -3 2 1 1 -4 -3 1 0.918 0.045 0.033 C B
K d h -1 -1 0 -1 -4 1 1 -2 -3 -2 -3 2 -2 -4 5 0 2 -5 -3 -3 0.909 0.050 0.040 C B
⋮
⋮
>1dbwa
M Z Z -6 -6 -7 -8 -6 -5 -7 -8 -7 -4 -3 -6 11 -5 -8 -6 -6 -6 -6 -4 0.966 0.021 0.013 C E
Q Z i -2 -4 3 -3 -4 1 0 -4 2 -5 -4 -3 -3 -6 -2 4 5 -6 -5 -4 0.943 0.019 0.034 C B
D d i -1 -2 1 3 -5 2 3 -2 -2 -4 -4 2 -3 -5 2 1 1 -5 -4 -4 0.940 0.014 0.047 C B
Y d j 2 -2 -3 -3 -2 2 -2 0 1 -3 -2 -2 -3 -2 5 -1 -1 -5 -2 -3 0.849 0.007 0.141 C B
T d i -2 0 -2 -2 0 -1 -3 -4 -3 2 2 -2 1 -3 0 -1 3 -2 -2 2 0.299 0.003 0.688 E B
⋮
⋮

Apéndice B

Herramientas usadas en los modelos evaluados

A continuación se describen las herramientas usadas en la implementación de los modelos utilizados en este trabajo:

B.0.1. Campo aleatorio condicional CRF

Los modelos CRFs implementados en este trabajo se realizaron con el software CRFsuite [52]. Implementa el modelo CRF lineal de una forma óptima en el uso de recursos computacionales y el tiempo de entrenamiento es menor comparado a otras herramientas. Las ventajas de CRFSuite sobre otras herramientas son: proporciona la opción de realizar validación cruzada y puede dar como resultado la precisión y sensibilidad en el conjunto de prueba; implementa varios algoritmos de optimización de parámetros; proporciona un formato de entrada que permite que se pueden tener número arbitrario de características por observación; tiene la opción de proporcionar pesos en las características (es útil cuando se tienen funciones características continuas).

B.0.2. Campo neuronal condicional CNF

Los modelos CNF implementados en este trabajo usan la implementación de [21]. Está escrito en c++ con mpi y utiliza LBFGS como procedimiento de optimización de parámetros. El código fue modificado para recibir un archivo de prueba e imprimir un archivo de salida con las probabilidades marginales para cada observación. Ya que el código original realiza una partición automáticamente del conjunto de entrenamiento de entrada para probar y no se puede realizar validación cruzada.

B.0.3. Red Neuronal

La implementación del modelo de dos capas donde cada capa corresponde a una red neuronal con una capa oculta. Se implementa con el *framework* de computación científica Torch. El *framework* usa el lenguaje Lua, con un gran conjunto de ricas características para la implementación de redes neuronales profundas.

B.0.4. SvmPrat

SvmPrat es un *framework* que permite construir modelos SVM para anotar residuos de aminoácidos en secuencias de proteínas con las características proporcionadas [56]. Implementa un modelo en cascada de dos capas, donde cada capa tiene K clasificadores binarios de uno versus el resto. Así es como en el problema de predicción de elementos con el alfabeto PB, el *framework* entrena 32 máquinas de soporte vectorial.